A DESIGN OF LOW POWER WEARABLE SYSTEM FOR PRE-FALL DETECTION

A Thesis

Submitted to the Faculty

of

Purdue University

by

Neeraj R. Rathi

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

May 2018

Purdue University

Indianapolis, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Maher E. Rizkalla, Chair

Department of Electrical and Computer Engineering

Dr. Mohamed A. El-Sharkawy

Department of Electrical and Computer Engineering

Dr. Paul Salama

Department of Electrical and Computer Engineering

Approved by:

Dr. Brian King

Head of the Graduate Program

I dedicate this work to my Mom and Dad for their endless support in making me successful acquiring my dreams.

ACKNOWLEDGMENTS

Firstly, I would like to express my profound gratitude to my advisor Prof. Maher Rizkalla for giving me research opportunity and guidance throughout the master program. His expert advice and motivation helped me excel in my studies and pioneer to my area of interest.

I am grateful to Prof. M. El-Sharkawy for teaching me the concept on embedded system and wireless communication. His classes Design with Embedded System, and Wireless and Multimedia Computation helped me tackle the difficult states of my thesis.

My sincere thanks to Nikhil Tiwari for giving me the insights of Pre-Fall detection and helping me understand his approach. I thank my fellow mates for their endless encouragement and support throughout my master's journey. I am also thankful to IUPUI and Dr. Brian King for offering me admission for master's program. Dr. King's sessions on selecting a major, and registering for an appropriate course helped me direct towards the right path.

Last but not the least; I thank my parents and my best friend Mrunal, for being patient and enthusiastic during my hard time. My incredible journey at IUPUI was a result of immense knowledge I gained from professors and staff, and valuable friendship from classmates.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure | Page

SYMBOLS

ω	omega
θ	theta
μ	micro
m	mili
A	ampere
t	time
V	volts
k	kilo
M	mega
n	nano
Hz	hertz

ABBREVIATIONS

MCU	Microcontroller
ADL	Activity of Daily Living
BLE	Bluetooth Low Energy
MEMS	Micro Electro-mechanical Systems
SVM	Signal Vector Magnitude
SMA	Signal Magnitude Area
API	Application Program Interface
ISR	Interrupt Service Routine
RAM	Random-access Memory
I2C	Inter-Integrated Circuit
GPIO	General Purpose Input Output
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
LED	Light Emitting Diode
ADC	Analog to Digital Converter
RF	Radio Frequency
GATT	Generic Attribute Profile
GAP	Generic Access Profile
AT	Attention(AT command)
TX	Transmit
RX	Receive
SDA	Serial Data Line
SCL	Serial Clock Line
DMP	Digital Motion Processor

ABSTRACT

Rathi, Neeraj R. M.S.E.C.E., Purdue University, May 2018. A Design of Low Power Wearable System for Pre-Fall Detection. Major Professor: Maher E. Rizkalla.

Fall in recent years have become a potential threat to elder generation. It occurs because of side effects of medication, lack of physical activities, limited vision, and poor mobility. Looking at the problems faced by people and cost of treatment after falling, it is of high importance to develop a system that will help in detecting the fall before it occurs. Over the year's, this has influenced researchers to pursue the development to automatic fall detection system. However, much of existing work achieved a hardware system to detect pre and post fall patterns, the existing systems deficient in achieving low power consumption, user-friendly hardware implementation and high precision. Growth in medical devices can be seen in recent years. Today's medical devices aim to increase the life expectancy and comfort of human being. The systems are designed to be made reliable by improving the performance, optimizing the size and minimizing the energy consumption. For wearable technologies, power consumption is an important factor to be considered during system design. High power consumption decreases the battery life, which leads to poor comfortability. The purpose of this research is to develop a system with low power consumption to detect human falls before they happen.

This research points towards the development of dependable and low power embedded system device with easy to wear capabilities and optimal sensor structure. In this work, we have developed a device using motion sensor to sense the subjects linear and angular velocity, communication sensor to send the fall related information to caretaker, and signal sensors to communicate and update user about device information. The designed system is triggered on interrupts from motion sensor. As soon as the system is triggered by an interrupt signal, users balanced and unbalanced states gets monitored. Once the unbalanced state is designated, the system signifies it as fall by setting a fall flag. The fall decision parameters; pitch, roll, complementary pitch, complementary roll, Signal Vector Magnitude (SVM), and Signal Magnitude Area (SMA) are layered to classify subject's different body posture. This helps the system to differentiate between activity of daily living (ADL) and fall. When the fall flag is set, the device sends important information like GPS location and fall type to caretaker. Early fall detection gives milliseconds of time to initiates the preventive measures.

The system was designed, developed, and constructed. Near 100% sensitivity, 96% accuracy, and 95% specificity for fall detection were measured. The system can detect Front, Back, Side and Stair fall with consumption of 100μA (650μA with BLE consumption) in deep sleep mode, 6.5mA in active mode with no fall, and 14.5mA, of which 8.5 mA is consumed via the BLE when fall is declared in active mode. The power consumption was reduced because the integrated wireless communication devices consumed power only when the fall is triggered, giving the device a potential to communicate wirelessly.

1. INTRODUCTION

1.1 Motivation

Falls in elder generation has become the frequent cause of injuries, physical disability, and death. Fall happens when people lose their balance state and enter into an unbalance state, which brings them to rest inadvertently on the floor or lower lever. According to World Health Origination (WHO) [1], fall is second leading cause of unintentional injury death after road accidents. Each year approximately 646000 individuals die from fall globally, in addition, 37.3 million falls are severe enough to require medical attention [1]. An average annual cost estimated on treating a fall accident exceeds \$30,000 [2]. Figure 1.1 shows the linear relation between the fall injuries and Medicare cost [3, 4]. As it indicated, in 2012 the cost of treating fatal and non-fatal injuries were estimated to be \$35 billion in U.S and assumes to increase to \$101 billion by the end of 2030 [2]. Regardless of extensive fall prevention programs [1], the number of falls and medical care is estimated to increase every year. Most importantly, un-expected fall become dangerous for a person living alone. Staying with fall injuries for a long time may lead to hypothermia, rhabdomyolysis or dehydration. Near 20% of people, suffer severe injuries like trauma or fractures [5] because they were unable to get help. Moreover, the injuries among elders may lead to a rapid decline in health and increases the chance of early death [6]. As the population is aging [7] the possibilities of fall among elderly, which is often due to chronic health conditions, dizziness, muscle weakness and illness is increasing. Aging is inevitable but fall detection and prevention should not. The study in [8] concludes that fall is an important factor of admittance into nursing home and therefore the system that prevent fall needs to be implemented to delay or reduce the chances to admittance into nursing home. The need of a device that finds preventive measure to

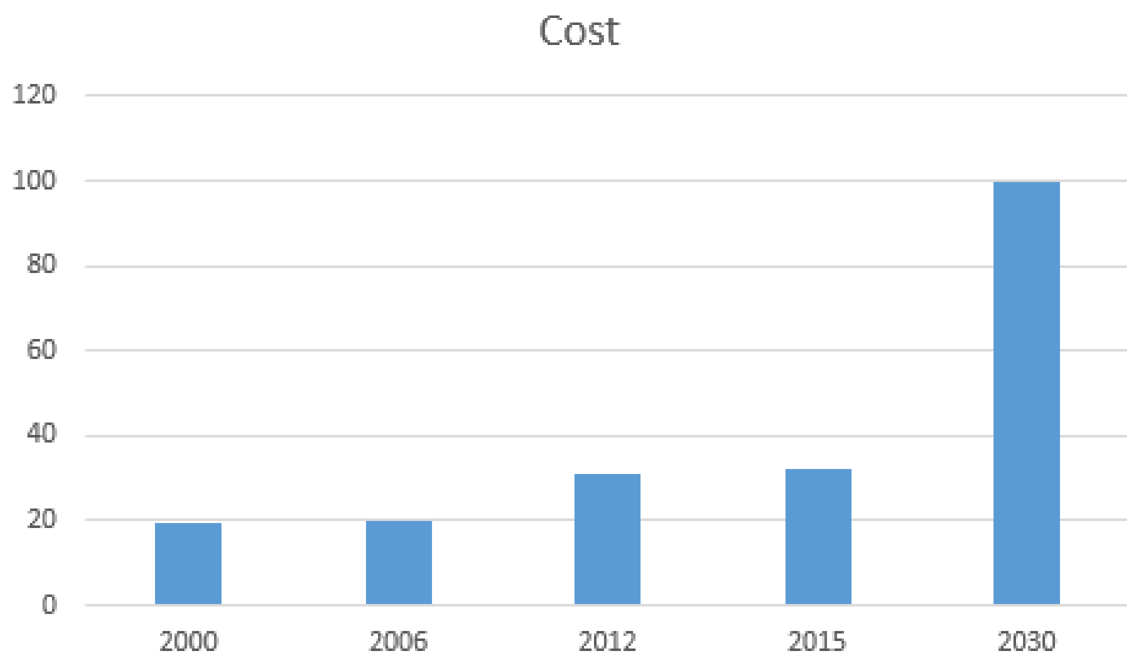

Fig. 1.1. Medicare Cost Data

minimize unexpected fall is necessary. The fall and its related injuries is a challenge in the medical field. A reliable system to detect early fall, deploy safety measures and communicate fall information with the caretaker is highly important to decrease the admittance ratio of elderly people into the hospital.

1.2 Fall Detection Devices

The importance of detecting and preventing a fall is unavoidable. Many researchers have also considered this issue and enormous work has been done to design a fall detection system with different approaches. In every related research, the authors have tried to overcome the existing research or proposed new technique to identify

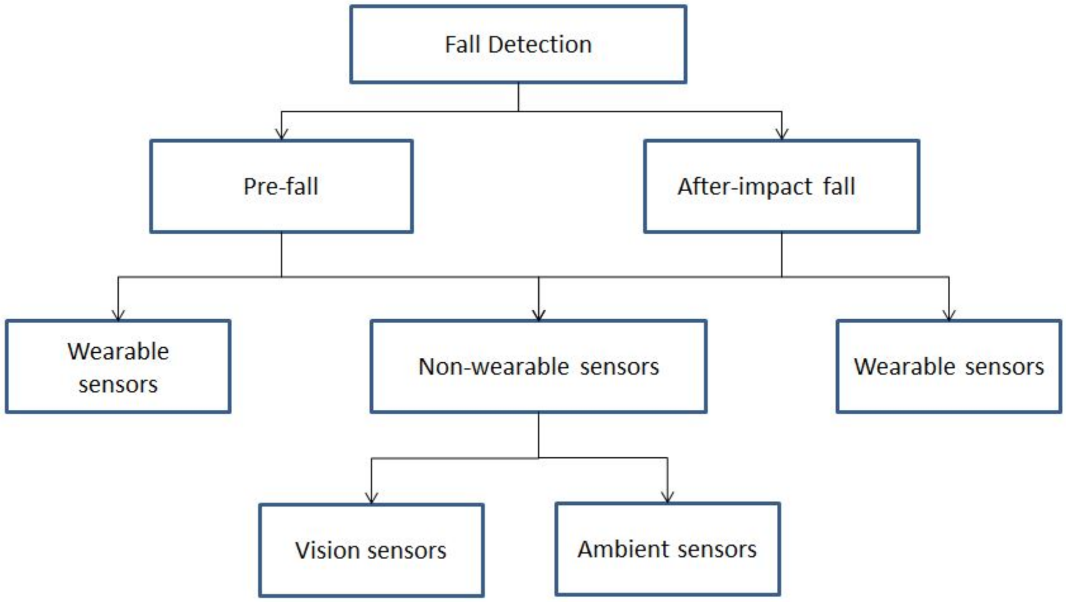

Fig. 1.2. The Design Architecture of Fall Detection Systems

falls. Researchers have worked to develop systems with different categories but the fall detection principles is either based on detecting fall before it happens know as Pre-Fall or detecting a fall after it has taken place known as after impact fall.

Many fall detection techniques found in literature are designed with both wearable devices and non-wearable devices. Figure 1.2 shows the design categories of the existing systems. The non-wearable devices are embedded into the surroundings to detect falls. It includes the combinations of sensors like floor sensors, infrared sensors, camera, microphones, pressure sensors, acoustic sensors, and location sensors for analyzing falls. The wearable device includes accelerometer, gyroscope, velocity sensors and mercury tilt switches to analyze fall. In this section, we will discuss the fall detection techniques based on Pre-Fall and after-impact fall detection.

1.2.1 After-impact Fall Detection Devices

In after-impact fall detection, the fall is detected after the subject has fallen. Figure 1.3 gives an example of after-impact fall detection with the use of accelerometers. The MCU (micro-controller unit) reads the real-time accelerometer raw values to track the user. If the raw value is less than 1g (gravity unit) on any of X, Y, Z axis the free fall is declared. Once the free-fall is declared then the device looks for high peak in the raw values, which comes when user hit ground after fall. If the peak is achieved the device then looks for inactivity of user, that is monitoring if user is subconscious after the fall. If the user is not performing a normal activity after the high peak is detected, then the fall is declared.

Adlian Jefiza [9] used Accelerometer, Gyroscope, and Back Propagation neural network approach to detect human fall. The author used MPU6050 with Ardunio Nano and SD card to process the data. The hardware is placed in a waist bag to make it easy to wear for the user. The Microcontroller takes the data from MPU6050 at 20Hz. The data is then used for preprocessing and feature extraction, where the data is normalized based on its value (minimum, maximum and average). After preprocessing, the back propagation is used to differentiate between fall and activity of daily living. It uses 3 layers BPNN (Backpropagation Neural Network) to get the optimal results. Layer 1 is input where the accelerometer and gyroscope data is received, while layer 2 includes the work pattern of BPNN, and layer 3 gives the expected output pattern. The layer 2 calculates the output value. If the desired and output value does not match then the weight in network is updated. The precision of 98.33% with the overall accuracy of 98.182% is obtained by this approach.

Smriti Bhandari [10] used vision-based approach to detect fall. The system developed in this research was integrated into the home to track the individuals. The camera was used to capture the video and provide video frames to the algorithm. The algorithm finds the interest points from the frames, and computes the optical flow to estimate the speed of motion. It then compares the last interest point with the

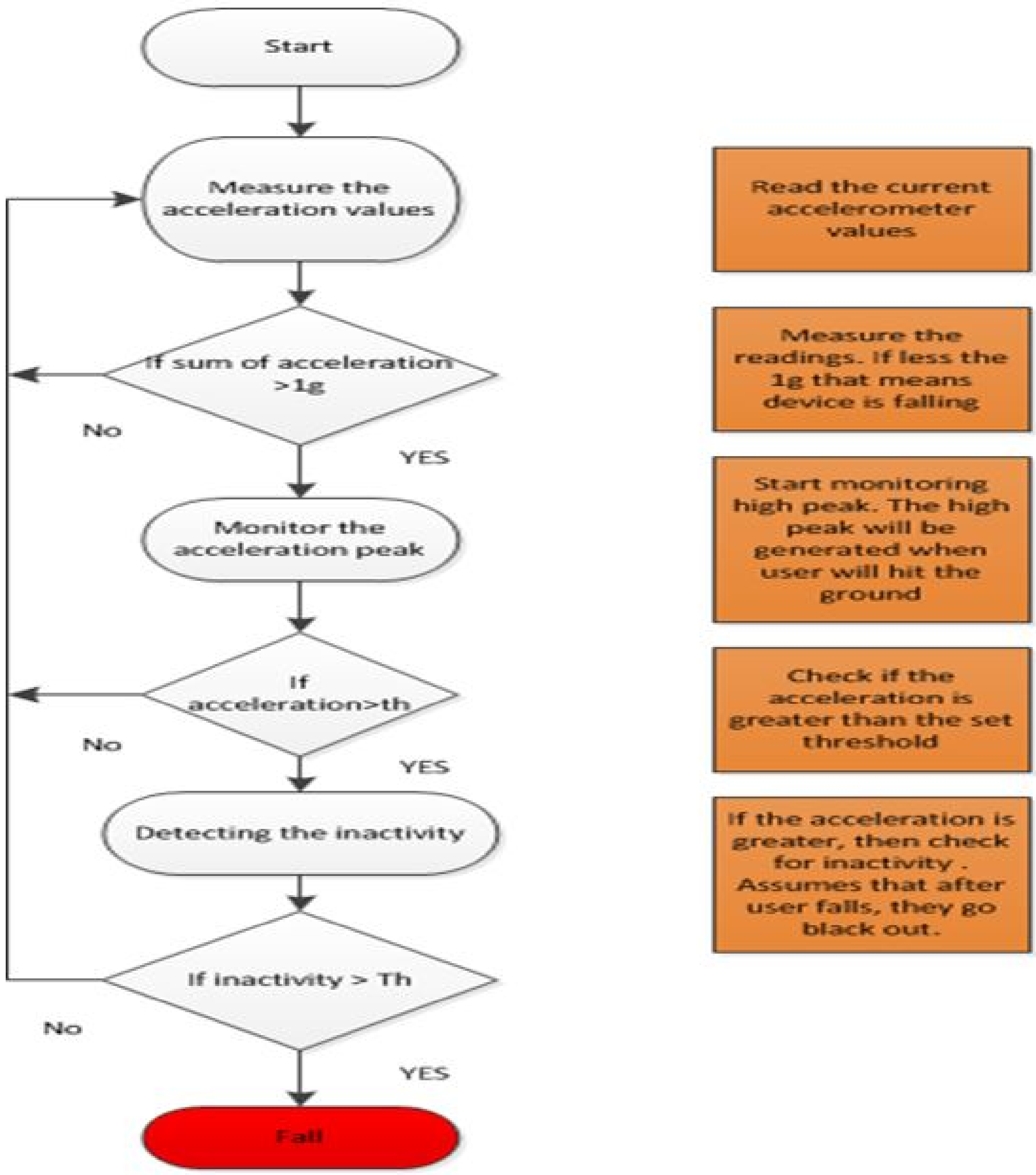

Fig. 1.3. Flow Diagram of After-impact Fall Detection System

new interest point to determine the rest position of user. The significant difference in speed of motion with downward direction and displacement greater than threshold, and current position near floor determines fall. The precision of 93% with the overall accuracy of 95% was obtained by this approach.

Mihail Popescu [11] used an acoustic fall detector system to detect fall. The fall detector system consists of microphones along with motion sensor to avoid false alarm. The system computes fall based on acoustics and uses motion sensor to cross verify by detecting motion during the fall interval. If the fall is detected and during the fall interval there is no motion then the system declares it as fall and sends a message to care taker.

Lucio Ciabattoni [12] used mobile robots to determine human fall. The system used smart phone and mobile robot to determine fall. The device first analyzes the accelerometer data from the smartphone online. Once the fall event is triggered, then the beacons sensor information is used to provide the location. The mobile robot moves towards the provided location to verify if the triggered event is a fall or not, through voice interaction with the fallen user. This approach achieved 99% sensitivity with 74% specificity.

A fall detection using wrist-worn accelerometer and barometer was proposed by Prayook Jatesiktat [13]. Fall is detected by integrating a three axial accelerometer and barometer on a wrist. The accelerometer (LIS3DSH) is sampled at 100 Hz with 6 g, and the barometer (LPS25HB) is sampled at 25 Hz with resolution of 24 bit. LPC54102 microprocessor is used to communicate with the sensors and process algorithm. The device captures the raw data from accelerometer and calculates the magnitude without applying filter. The barometer raw values are captured and filtering is applied to reduce the noise. Once the accelerometer values reaches to 4g then the algorithm traces for free fall event. Free fall event reduces the magnitude to 1g when the subject hit the ground. The pressure data from barometer is used for feature extraction: Pleasure shift, Middle slope and Post fall slope. This data helps to signify fall when combined with free-fall.

Work proposed in [14] [15] [16] uses smart phones for detecting fall. The approach [14] used accelerometer within the smart phones to capture the subjects posture information such as lying, sitting and standing without movement, horizontal movement, vertical movement, and fall. The device uses smart phone messaging or multimedia messaging to send the fall related information to caretaker. Rahul Tiwari [16] used android platform for fall detection and proposed the touch screen alternative to reduce the false triggered fall. Frank Sposaro [15] continued the Tiwaris work and provided a feature of sending SMS to caretaker and putting the phone on speaker to confirm the fall identity when caretaker responses.

There is lot of work done for the non-wearable solutions. The research in [17] [18] [19] make use of vision sensors, including depth cameras. In [20] radar was used as an ambient sensor to detect falls. Solution based on pressure-sensitivity fibers to develop smart embedded floor have been proposed in [21].

1.2.2 Pre-Fall Detection Devices

In Pre-Fall detection, the transition from balanced to unbalance state is monitored. Once the subject reaches the unbalance state i.e. the state from where we cannot return to balance state the fall is detected. The example in figure 1.4 below shows the Pre-Fall detection algorithm flow diagram for accelerometer based device. The device takes the accelerometer raw data and filters it to reduce the noise. The filtered values are then compared to the threshold to determine if the user motion is above the set limits. If greater, the fall parameters are calculated to determine fall or activity of daily living (ADL). The fall is declared once it is confirmed that the values representing motion is not ADL. Nuth Otanasap [22] implemented this approach. The accelerometer information is shared with smartphone, via Bluetooth. The author used nRF51822-based board; build with Arm Cortex M0 and Bluetooth. This board is attached at chest location to track the velocity and acceleration characteristics.

The smartphone is used as data monitoring unit, which runs the algorithm and takes the fall decision. The developed device obtained 99.48% sensitivity, 95.31% specificity and 97.40% accuracy.

Jian Liu [23] have proposed a prior-to-impact fall detection algorithm using2D-information (i.e. trunk velocity and trunk angle). They proposed to use unexpected slip-induced fall for the validation and development of algorithm. The experimental process consists of one IMU placed at sternum to measure the linear and angular velocity of subject's body in 3-D at sampling frequency of 100Hz. The system requirement also consists of a sensing system with six infrared motion capture camera (ProReex MCU 240, Qualisys) to calculate the 3-D positions of reflective markers at 100 HZ sampling rate. In this algorithm, first, the IMU signals are captured and trunk extension angle and angular velocity were measured. The values of trunk extension angle and angular velocity are then given to fall discriminant function to provide a differentiation between ADL and fall. Once it is classified as fall then the threshold is used to make the final decision.

Alessandro Leone [24] proposed a Pre-Fall detection system based on Electromyography (EMG). The system identifies highly discriminative features extracted within the EMG signals for instability detection. The threshold-based approach is used to detect an imbalance condition about 200ms after the stimulus perturbation, in simulated and controlled fall conditions. The hardware system uses wearable surface EMG FREEEMG1000 developed by BTS Bioengineering. The wireless probes are used on user's body to capture the data. The Probes sends the received data to a stand-alone PC running data acquisition and elaboration algorithms.

Shaoming Shan [25] implemented Pre-Fall detection algorithm with the machine learning approach. The author's objective was to detect the Pre-Fall with wearable device but due to high computing time and more hardware requirements that made this system too complex to build. The author has used STMicroelectronics LIS3LV02QD accelerometer to calculate the body acceleration and NEC 78K0547

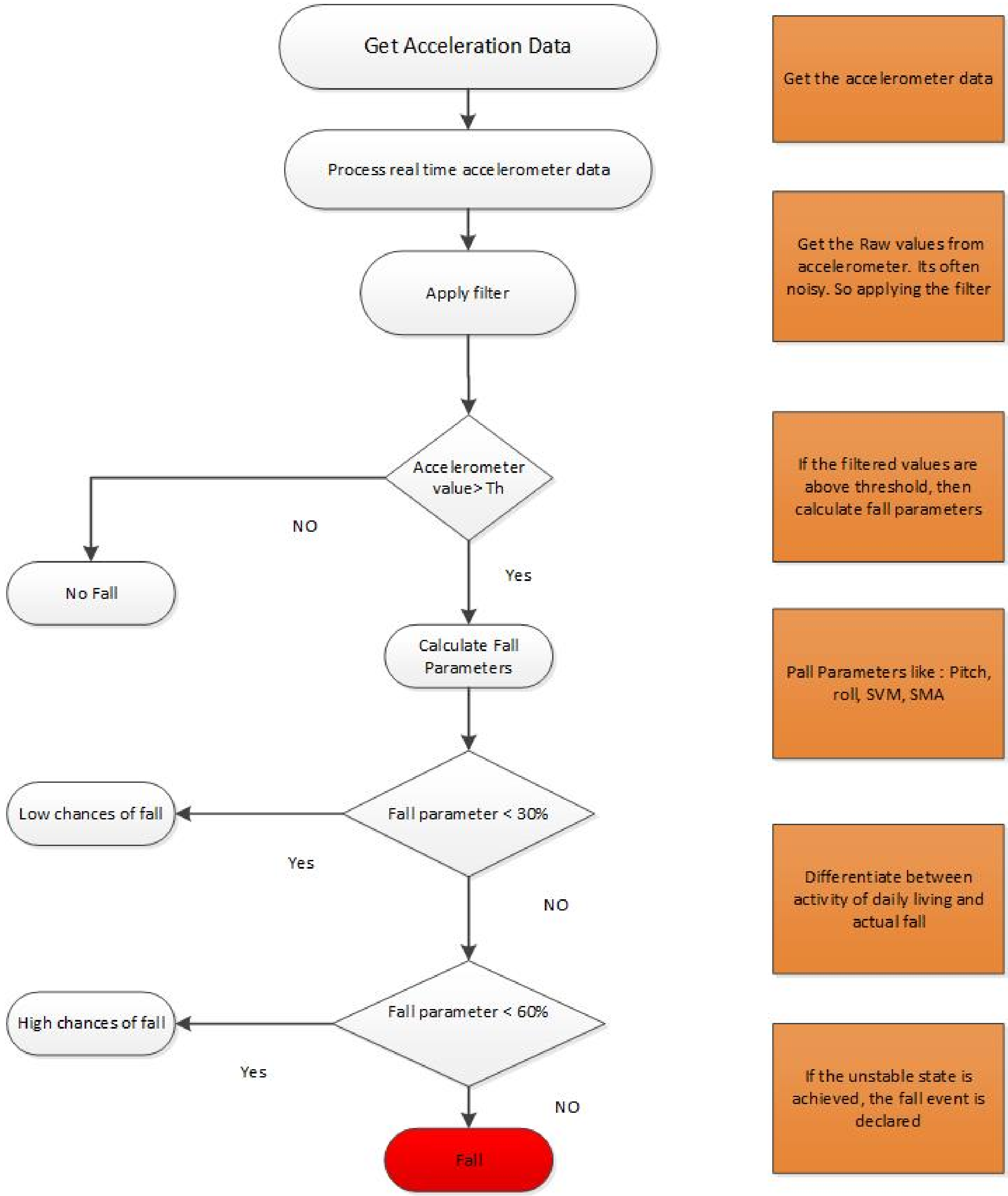

Fig. 1.4. Flow Diagram of Pre-Fall detection system

Micro-controller Unit (MCU) to process the data. The MCU collects the data from the accelerometer and transmits it to the PC using nRF2401 wireless module. The PC computes the fall algorithm and takes the decision.

Ge Wu [26] have proposed the pre-impact fall detection system with inertial sensors. The developed system is portable and detects fall before it occurs. The sensor used for measurement consists of tri-axial accelerometer and a tri-axial gyroscope. Pocket PC (HP iPAQ h5550) does processing of the sensor data. The pocket PC can log 20 hours of data at 57 HZ. The complete system is placed in a waist bag like encasing.

1.3 Issues and Limitations of Existing Devices

The approaches discussed above show research in pre and post fall detections. In recent years, it has progressed from analyzing correct fall pattern to implementation of new hardware fall detection tech (Smartwatch, smartphone). The designed approaches have several advantages over one another, but all majorly lack in meeting low power requirements and standalone system, in addition, they are complex. Energy consumption is very important factor to be considered when designing a portable or wearable device. High-energy consumption increases cost and decreases device efficiency and comfort.

The Pre-Fall detection system proposed by Ge Wu [26] and Jian Liu [23] was based on threshold-based fall detection using inertial sensors. The system works by continuous monitoring of user activity and logging the data. Such utilization of CPU results in high power consumption and reduces the comfortability of portable device due to poor battery life. Also, fall decision based on single sensor often results in high false alarms. Shaoming Shan [25] approached for Pre-Fall detection using machine learning. The system was contentiously measuring data and transmitting it to PC to provide fall information. The transmission of data to PC was done wireless hence compromising battery life by huge factor. In addition, the portability of system was a concern because of the complex machine learning algorithms. The method proposed by Alessandro Leone [24] uses probes at different body location making it uncomfortable to use on daily basis. In addition, the device is not standalone; it requires external PC to process data.

In [9-21] fall detection devices detects fall once it has taken place. Such system can only inform about fall, but will not be able to take any preventive measure. On fall, these devices either send message to care taker or make users to press a button to request emergency. [14-16] used smart phones to detect fall detection. The approach successfully reduces the hardware requirements but decreases the efficiency. Smart phones have its own purpose and integration of dependent application sometimes give false data. Considering the fall situation when user is on call, smartphone is put up for charging, and kept aside while sleeping.

Apart from the portable system, the non-wearable system described in [10] [17-21] has issue with cost, range, and efficiency. The vision-based system needs integration into surrounding which compromises the security, privacy and range. On the other hand, acoustic based approach provides less cost but has a noise issues. On contrary to non- wearable devices, the wearable devices developed using Micro-Electro-Mechanical Systems have advantage of being available easily, less computation, smooth integration, and cheap. While using the wearable sensors, it is of importance to place them at right location on body. The body location of a sensor affects the detection capabilities [27] [28] . From [29] placing IMU on waist results on best accuracy in fall detection.

Table 1.1 gives the overview of current consumption of microcontroller and development boards used in the past for fall detections. The current consumption listed is not derived from the research listed because of the lack of data provide in the related studies. The data was obtained from the hardware specification data sheet.

In summary, there is need for a system that consumes less energy to be more appropriate for wireless in addition to being a friendly user. The proposed system here addresses these important issues. The power consumption, the hardware/software

Table 1.1.
Summary of power consumption in Portable fall detection systems.

Sr.No	Research	Microcontroller	Approx. Current consumption (MCU)	Method
1	[13]	LPC54102	9mA [13]	Polling
2	[30], [31]	Atmega328P	15-20mA	Polling
3	[28]	raspberry pi 3	0.6A	Polling
4	[26]	3DM	50mA	Polling

complexity, and above all they provide better safety for patients. In addition, the existing systems do not examine how the software and hardware can be integrated into a wearable device in an efficient and non-intrusive manner, while this issue was tackled in our approach. Furthermore, the designed system concerns the accuracy and specificity. The system implemented here optimizes the number of sensors and their locations on the human body to be appropriate for use and production.

1.4 Employment of Low power in Wearable Embedded Devices

Wearable embedded system devices are often small, portable, and hand-held. It is an integration of sensors with one or many control units to process the sensor data and execute the results based on the information received. It is important to consider performance, power, flexibility, cost and reliability of portable systems as the requirements of future devices with advanced features are getting popular. The performance of device plays a vital role in defining the application it is used in. The efficient software is important towards the speed of the device. The clock cycle and instructions executed determines the speed performance of the system. The use of complex processors to achieve the system performance increases power dissipation and therefore, suitable measures need to be taken to minimize the power consumption in wearable and battery-operated devices.

The power efficiency of a device is achieved by reducing power dissipation. The system hardware for all stages of the device process need to be designed without compromising system performance, size, and cost. As power is important factor in design, it needs to be addressed carefully to meet performance requirement. The decrease in the size of the system and increase in the integration level results in more heat generation and hence increases power dissipation. Therefore, it is a challenge to design a device by suitably selecting the advantages and disadvantages of various processors, ICs, and reliable hardware and software integration. The tradeoff between energy consumption, improved performance, size and cost can be handled by considering different sets of design choices while implementation.

In electronic systems, it is understood that managing the average power dissipation will minimize the manufacturing and packaging cost with increased reliability. Power reduction can be analyzed by identifying and optimizing the important factors affecting the power dissipation. Power dissipation contributes to dynamic power and static power.

$$PowerDissipation = DynamicPower + StaticPower \tag{1.1}$$

Static power: In submicron technologies, static power dissipation is a result of leakage current and subthreshold current contributing to small percentage of total power consumption. As the circuits scaled down in size the static power consumption increases.

Dynamic power: A changes of state when device is active results in dynamic power dissipation. Dynamic power is also referred as sum of switching and short circuit power dissipation. The extensive switching causes output loading resulting in 85-90% total power consumption due to dynamic power dissipation.

Power dissipation is major component in achieving low power consumption. Therefore, to minimize power dissipation, power reduction techniques for dynamic and static power needs to be implemented. These techniques can be applied at all level of design hierarchy, like Algorithm, Architectural, circuit design and device technology. As the power reduction techniques to reduce static and dynamic power can be applied at the Micro-controller level. It is very important to select a correct MCU and write optimal software.

Basis of MCU Selection

- Does the MCU supports low power Modes?
- If yes, the power consumption in active (Run) mode?

- Power consumption in sleep, deep sleep and standby modes?
- What is the wake-up time between the modes?
- What are the wake-up sources?
- Operating speed, Memory, package, GPIOs, and Peripherals.

Basis of Software Design

- Employing the device operating at different voltage.
- Employing a device operating at different frequencies.
- Implementing low power modes 1. Sleep, 2. Active, 3. Halt, 4. Stop.
- Employing a device capable of operating at different clock speeds.
- Employing scheduling technique in software to reduce power consumption.

Low power system design is a holistic process that is enabled by selecting a combination of proper devices, software, and development tools.

1.5 Our Approach

Pre-Fall detection system in this research is designed to overcome the existing work in this field. The development of Pre-Fall detection system is done by considering real life scenarios, user flexibility and one's comfort towards using the wearable devices. The designed system completely integrates on a belt like structure, which can be easily wore around the waist area. As modern day embedded system must consume less power, provide high dependability, efficient communication, and low cost, the proposed system helps achieving the following:

- **Automation:** The developed device to detect fall event without any user interaction.

- **Reliability:** Distinguish fall event with other human daily activities.

- **Power Consumption:** Developed device switches between power modes to save on power.

- **Low power Wireless:** Integrated with BLE capabilities to transmit fall information to care taker.

- **Reaction time:** The system detects Pre-Fall event approx. 250ms.

This work proposes a wearable sensor based on pre fall detection system, which uses linear and angular velocity information from motion sensor to classify human fall. The sensors are integrated along with the controller on to a user's belt. The fall alert is shared to an android application developed for receiving fall information from measuring system.

Research is also done to make sensor arranged in such a fashion that it gives high accurate data of human orientation with easy to wear on capabilities. The developed system also focuses toward the battery life of device with the designed algorithm to help minimize it. Figure 1.5 shows the various design methods to be used to design fall algorithm. The approach followed here is designated by the low power modes with interrupts. This approach is based on interrupts but utilizes microcontroller energy modes and low power peripherals. The designed algorithm disables the unwanted peripherals and clocks and switches operating frequency as we they move from one mode to another.

Design of a low power approach for Pre-Fall detection consists of selecting an ultra-low power MCU and low power sensors, and developing an optimal algorithm. The developed algorithm utilizes the MCU and sensor specification to achieve proper performance, efficiency, reliability, and less power dissipation.

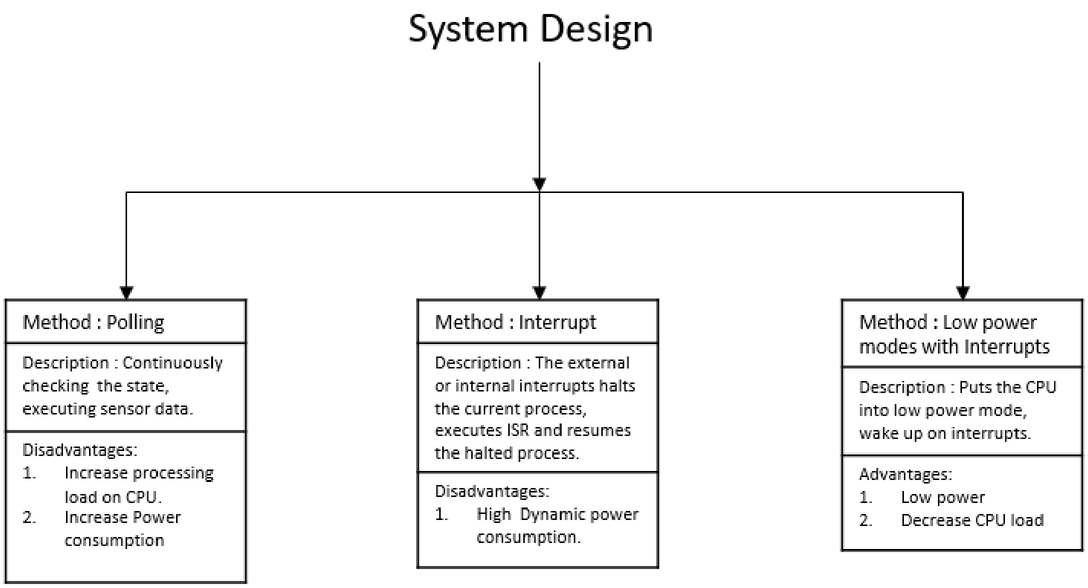

Fig. 1.5. The System Approach

1.6 Thesis Organization

The Thesis is organized in 8 chapters. As chapter 1 being introduction, chapter 2 cover the hardware components used in this work. The hardware section is further classified into control unit, motion sensor, and wireless sensor. The chapter 3 is communication protocol. This chapter gives the overview of the serial and wireless communication protocol used for data transfer between MCU and sensors, and MCU and Mobile application. The chapter 4 details the hardware and software design. It covers the hardware architecture, pin-to-pin mapping, and software flow diagram. This chapter also explains the Pre-Fall detection methodology. The chapter 5 explains the computational model build to estimate sensor information and calibrate then. It also details the fall parameter considered for Pre-Fall detection. The chapter 6 explains the low power approach and battery life estimation.

The Results are detailed in chapter 7. It explains the data collection, BLE mobile application, low power and fall detection pattern results. The research is then concluded with future proposed work in chapter 8.

2. HARDWARE

This chapter emphasizes the hardware components used for Pre-Fall detection. This include the micro-controller, and the sensor devices.

2.1 Control Unit

One of the important components to consider while developing an application is the selection of control unit (MCU). The MCU with sensor devices should feature high speed and low power for high performance system. These features will be detailed in later chapters. Table 2.1 gives the comparison between three popular ultra-low power micro-controllers. The EFM32GG was chosen for the design of power optimization of the Pre-Fall system. The MCU consist the following factors:

- Minimized active and sleep mode power consumption.
- Minimized processing time.
- Quick wake up time.
- Very low current consumption: 900nA in Deep sleep and 20nA in shutoff.
- Peripheral reflex system.
- Low energy Peripherals.

2.1.1 EFM32GG-STK3700 Development Board

EFM32GG-STK3700 is a development board for EFM32GG based microcontrollers from Silicon Labs. This development board is built around EFM32GG990F1024 MCU with 1 MB Flash and 128 KB RAM. The figure 2.1 shows the STK3700 devel-

Table 2.1.
Micro-controller comparison leading to the proper selection.

MCU Specification	**EFM32GG**	**MSP430F5**	**KL**
Architecture	32 bit ARM	120-25 MHz	48MHZ
Clock	48 MHZ	Atmega328P	15-20mA
Current : Active Mode	150μA / MHZ	200 μA / MHZ	5.7mA
Wakeup Time	2μs	3.5 μs	4μs
I2C	2	2	2
SPI	3	4	2
UART	3	2	2
RAM and FLASH(KB)	128-1024	8-64	32-265
Peripherals operating in power mode	RTC,LEUART,I2C, WDOG,LCD, I/O	I/O	LPUART,SPI,I2C ADC,DAC
Supply Voltage	1.98V-3.8V	1.8V-3.6V	1.7V-3.6V
Software Tools	Simplicity studio Power estimator	Code Composer MSP Energy trace	MCUXpresso
Development Boards	EFM32GG-STK3700	BoosterPack MKII	FRDM KL46Z

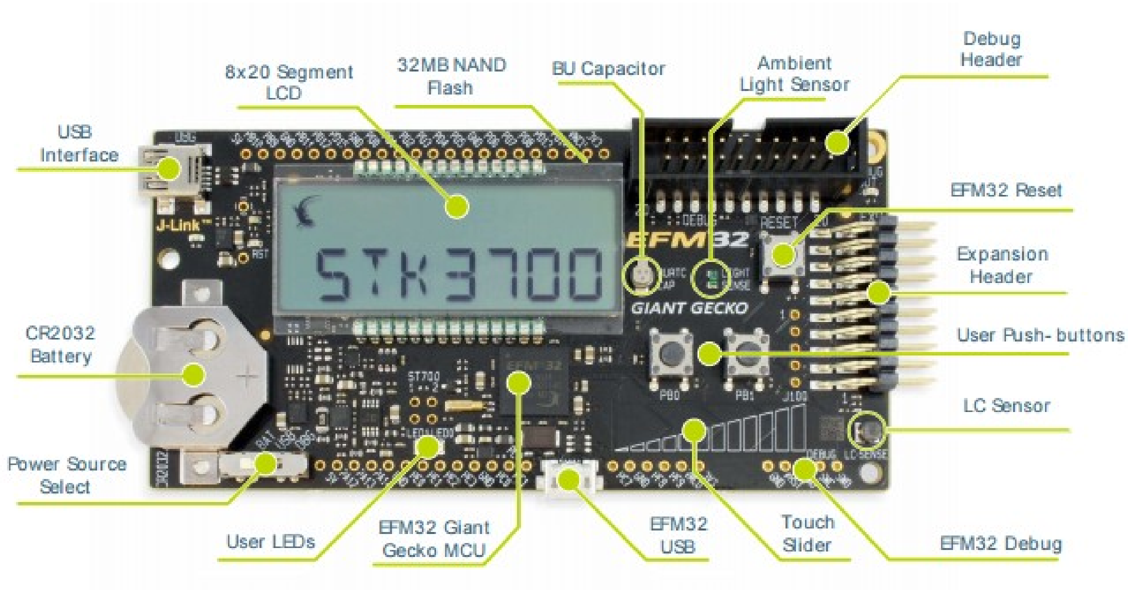

Fig. 2.1. The STK3700 Development Kit [32]

opment kit. This board contains sensors and peripherals to be used for application development and to demonstrate the MCU capabilities. The board features on-board SEGGER J-Link debugger and an Advanced Energy Monitoring system; to allow flashing the code, debugging and performing real-time current profiling of an application. The board displays many great features, which makes it the best choice of selection over other development boards. A few features are detailed below:

- 160-segment LCD Display.
- 0.03F Super Capacitor for backup domain capacitor.
- 32 MB NAND Flash.
- LFXO and HFXO: 32.768kHz and 48.000MHz.

2.1.2 EFM32GG Microcontroller

EFM32 MCU is a unique combination of the powerful 32-bit ARM Cortex-M3 with low energy techniques, short wake-up time from energy saving modes, and a wide selection of peripherals. These combinations make EFM32 the world's most energy friendly MCU in the market [33]. The EFM32GG microcontroller provides high performance and low-energy consumption; therefore, it is well suited for this battery operated application. This MCU is ideal for any application requiring flash memory configurations up to 1024 kB, 128 kB of RAM and CPU speeds up to 48 MHz. EFM32 contains two high frequency(HF) clocks oscillators, HFRCO and HFXO to source HF clock, and two low frequency clock oscillators, LFXO and LFCRO to run low frequency peripherals. Figure 2.2 shows the block diagram of EFM32GG MCU. The MCU includes the Core and Memory unit, Clock Management unit, Energy Management Unit, Serial Interfaces, I/O ports, Timers and Triggers, and Analog interfaces. The MCU also contains the peripheral reflex system, which let all the peripheral modules communicate with one another without CPU intervention.

The other EFM32 MCU features include the following:

- Low power sensor interface (LESENSE) with deep sleep monitoring.
- Energy management system with five flexible energy modes. The current consumption in each mode is,
  - EM0-Run Mode : 150μA/MHz at 3V.
  - EM1-Sleep Mode : 45μA/MHz at 3V.
  - EM2-Deep Sleep Mode : 1.1μA at 3V.
  - EM3-Stop Mode: 0.8μA at 3V.
  - EM4-Shutoff Mode: 20nA at 3V.

The MCU also features autonomous and low energy peripherals, AES encryption, pulse counter, low energy UART and sensor interface, and on-chip operational amplifiers.

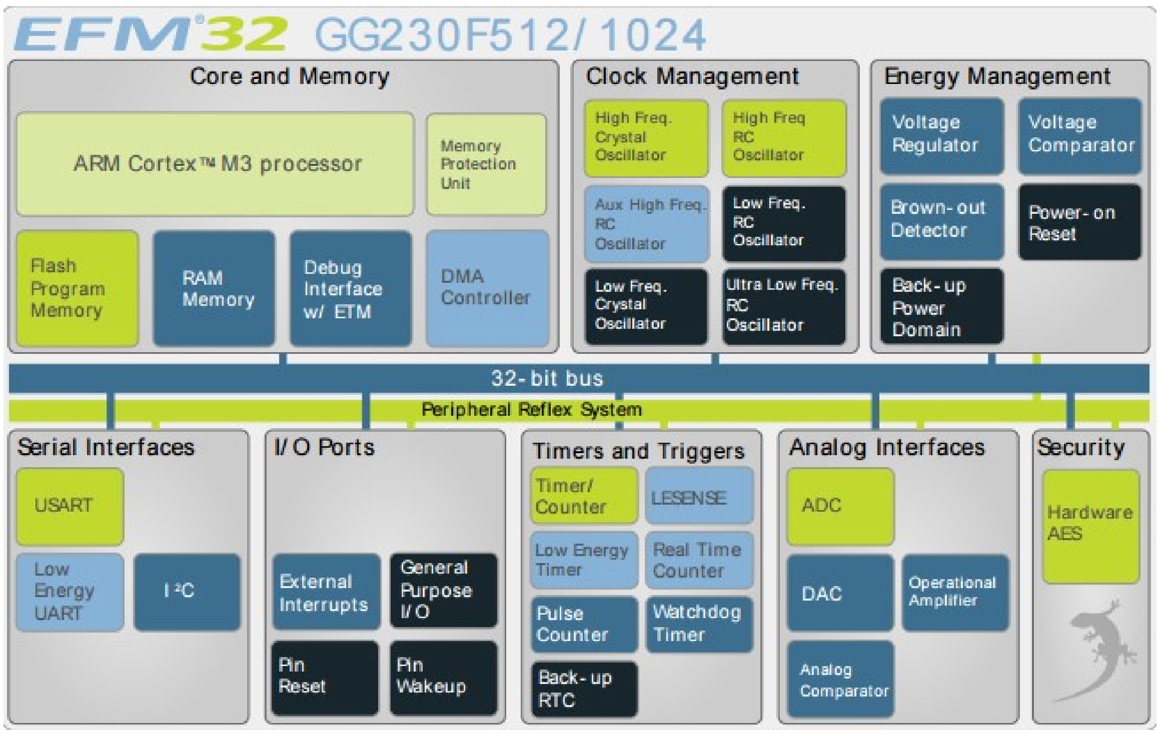

Fig. 2.2. The EFM32GG Architecture [33]

2.2 Motion Sensors

Applications in embedded world are developed based on the information it gets from the surroundings. The sensors measures the detected events or change in environment and informs to master device to make decisions. The sensors measures the physical quantities and converts it to digital, so the electronic circuits understand the information it is receiving. Sensor technologies have played a crucial role in transforming home automation, automobile, aviation, medical devices, and manufacturing. Therefore, sensors in todays embedded systems need to be accurately fabricate with proper electrical and mechanical standpoint. Power consumption and size of a sensor also plays an important role, when developing a portable battery power application. Recent advances in sensor fabrication solved power and size problem by increasing the transistor ratio on a micro with less spacing.

Motion sensing sensors are widely used where knowing orientation and acceleration is priority for their operation. In Pre-Fall detection system, the decision making is based on measurement from the motion sensor data. It requires information such as position, acceleration, pressure, altitude and orientation to identify subject's posture or orientation. For Pre-Fall detection conducted in this research, motion sensors like accelerometers and gyroscopes are used.

2.2.1 InvenSense MPU6050

The MPU-6050 is the integrated 6-axis Motion tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a digital motion processor (DMP) into a single 4x4x0.9mm package [34]. The MPU-6050 consists of three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and three 16-bit ADCs for digitizing the accelerometer outputs. The gyroscope in MPU6050 can be configured to user-programmable full-scale range of ±250, ±500, ±1000, and ±2000 degree/sec (dps) and the accelerometer can be configured to user-programmable full-scale range of ±2g, ±4g, ±8g, and ±16g for precision tracking of both fast and slow

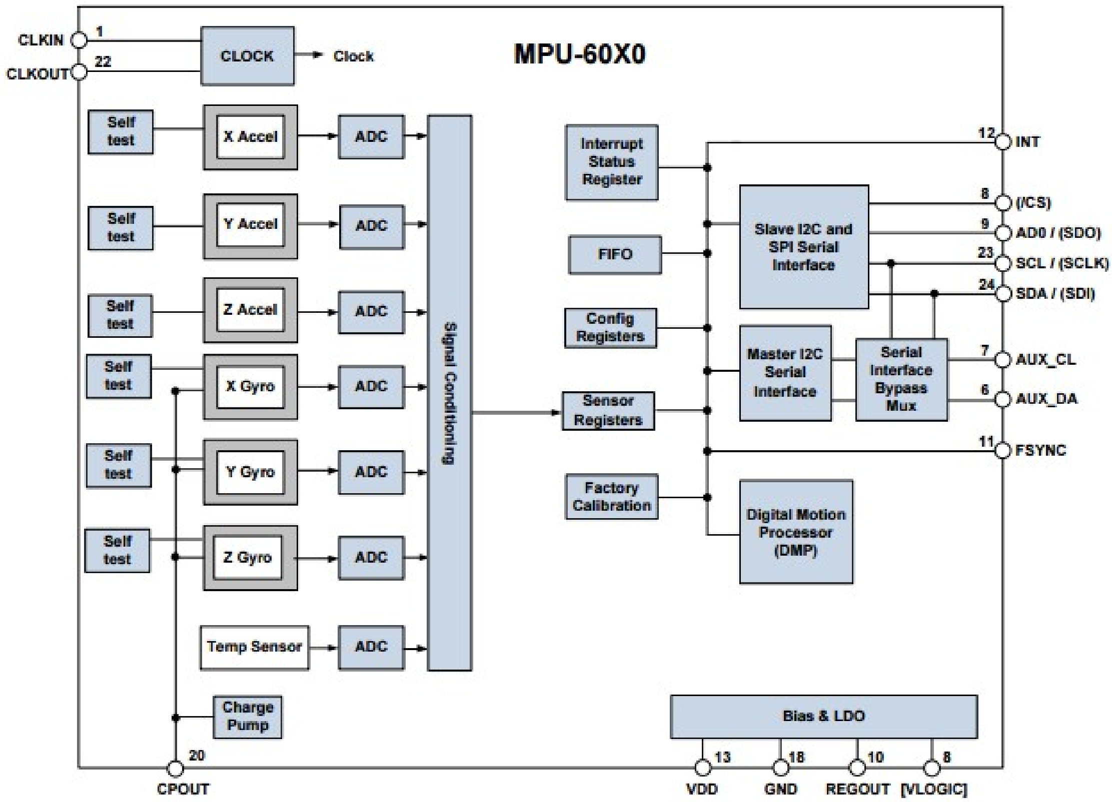

Fig. 2.3. The MPU6050 Functional Block Diagram [34]

motions. The chip feature 1024 byte FIFO buffer and low power mode to help lower system power consumption. The low power can be achieved by putting MPU6050 to sleep mode. The accelerometer remains active while gyroscope and DMP are disabled in sleep mode. MPU6050 supports only I2C interface to communicate with the MCU. It also supports user-programmable digital filters for gyroscope, and accelerometer. MPU6050 interrupts can be configured using user-programmable registers `INT_ENABLE`. Figure 2.3 shows the functional block diagram of MPU6050 [34].

The MPU6050 features in our design include:

- Configuring Clock.
- Enabling Low Power Mode.

- Setting Sensor Data Registers.
- FIFO.
- Gyroscope and Accelerometer Self-test.
- Reading Accelerometer and Gyroscope raw values.
- Bias and LDO.

2.2.2 Analog Devices ADXL345

ADXL345 is a 3-axial accelerometer with full output resolution of 13 bit at ± 16g by maintaining 4 mg/LSB scale factor in all g ranges. The high resolution enables the measurement of tilt/bent changes less than 1.0 degree with measuring static acceleration of gravity in tilt-sensing applications and dynamic acceleration resulting from motion or shock. It is also capable of working in ultralow power mode that enables intelligent motion-based power management with threshold sensing and active acceleration. The ADXl345 consumes as low as 40μA in low power mode with the sample frequency of 12.5HZ and 0.1 μA in standby mode. It can be interfaced with MCU using SPI (Serial Peripheral Interface) or I2C. It also supports software configurable interrupts: Tap/double tap detection, activity/inactivity, and monitoring free-fall detection by setting `INT_ENABLE` register. ADXL345 also supports user-programmable full-scale range of ±2g, ±4g,±8g, and ±16g with user-programmable data rate from 3200 to 6.25 Hz. Figure 2.4 shows the functional block diagram of ADXL345 [35]. The ADXL345 features in our design include:

- Configuring Clock.
- Enabling Low Power Mode.
- Setting Sensor Data Registers.
- Interrupts.

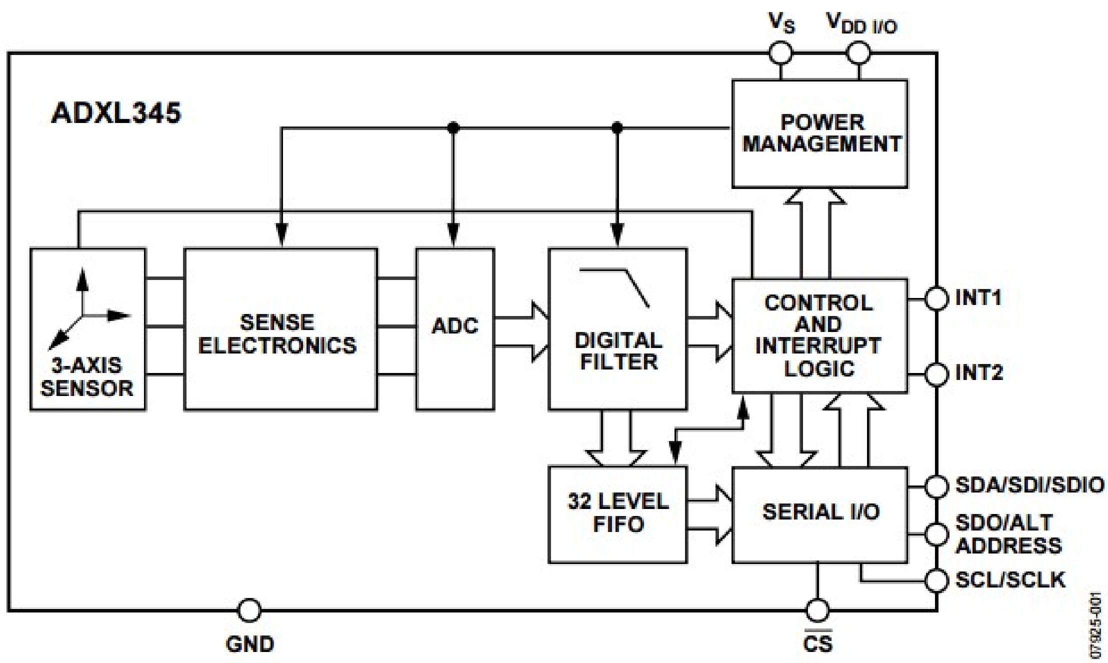

Fig. 2.4. The ADXL345 Functional Block Diagram [35]

2.3 Wireless Sensor HM-10

The wireless sensor is used to share fall related information with the care-taker. It gives us flexibility to connect devices without having a physical connection between them. HM-10, a low energy Bluetooth sensor is used to connect wirelessly and consume less power. It is breakout board with Texas Instruments CC2541 IC integrated on it. BLE is also known as Bluetooth smart as it brings reduced power consumption and cost in the similar communication range as compared to classic Bluetooth. The HM-10 uses BLE protocol to communicate with other BLE devices. It works on 2.4GHz ISM frequency band and uses maximum data rate of 2000 kbps. HM-10 has connection range of 100 meters and working temperature range of -5 to +65 degree centigrade. It switches between sleep and active mode when given desired AT commands. The HM-10 communicates with MCU using UART interface. The CC2541 SOC combines 8051 MCU with industry leading RF transceiver. It contains 8-kB RAM, programmable memory, and user-programmable output power. The Figure 2.5 shows the Functional Block Diagram of CC2541.

The HM-10 features in our design include:

- Configuring Baud Rate.
- Enabling Low Power Mode :- Sleep Mode.
- AT Commands.

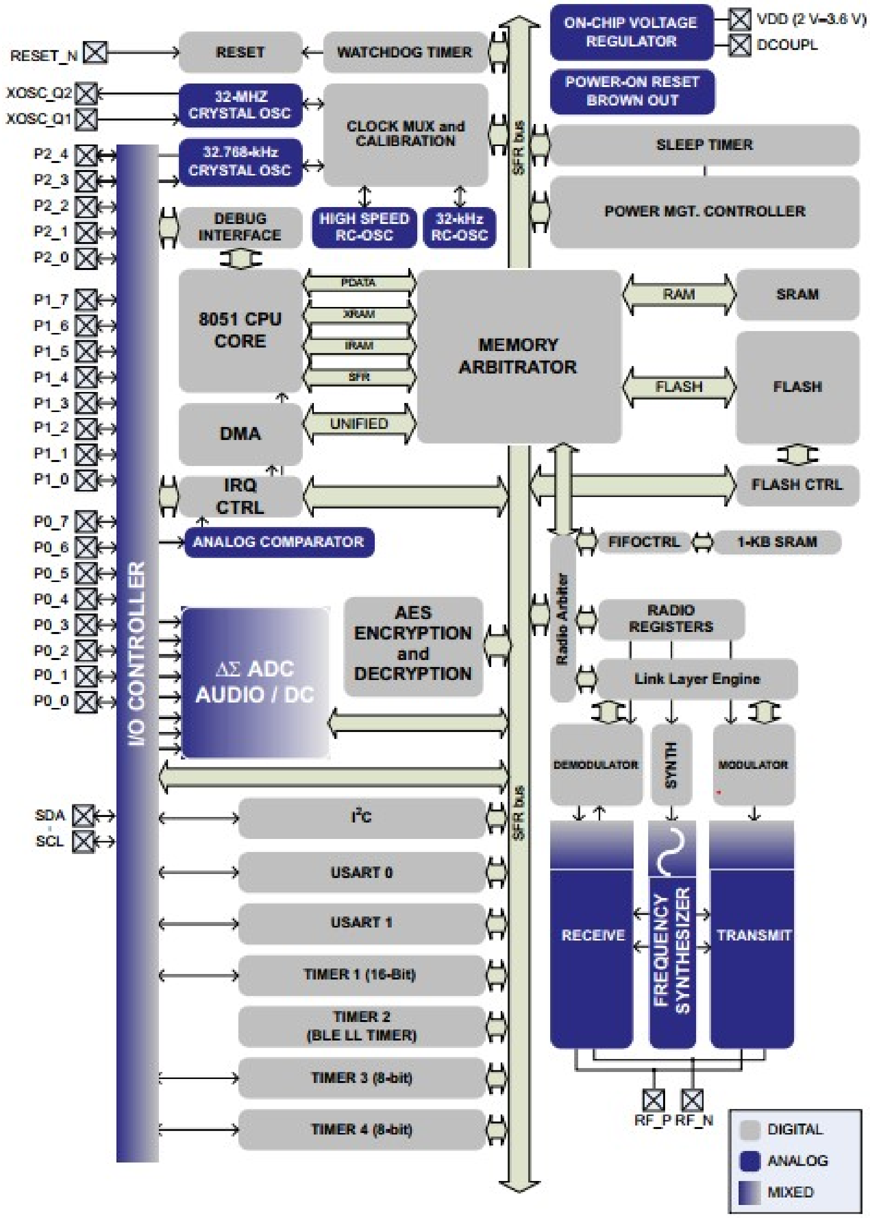

Fig. 2.5. The CC2541 Functional Block Diagram [36]

3. COMMUNICATION PROTOCOLS

Communication protocols are essential for an embedded system application. It helps in transmitting, and receiving of data from the sensors, MCU's, and devices. Below are the communication protocols used in this work.

3.1 Serial Communication Protocol

A single embedded system device contains different Integrated Circuits (IC). Therefore, it is often necessary to share information between them to create an application. For example, when we hit the acceleration paddle while driving, we see that the speed is increasing and at the same time, it is displayed on the dash board. Here, the paddle information is carried to motor and displayed using communication protocols. In serial communication protocol the data is transmitted one bit at a time, sequentially, requiring less wires compare to parallel communication protocol. The choice is obvious to obtain simple communication between IC while using less input and output pins, serial communication is highly preferred. For Pre-Fall detection system following serial communication protocols were used:

1. **Inter Integrated Communication (I2C):** I2C is widely used two wire serial communication protocol. Communication between two devices is established using Master-Slave combination. Bus in I2C contains two signals of SCL (clock), and SDA (data). I2C communication starts by setting SDA low while SCL high. Devices with I2C can communicate with the speed of 100 kHz or 400 kHz.

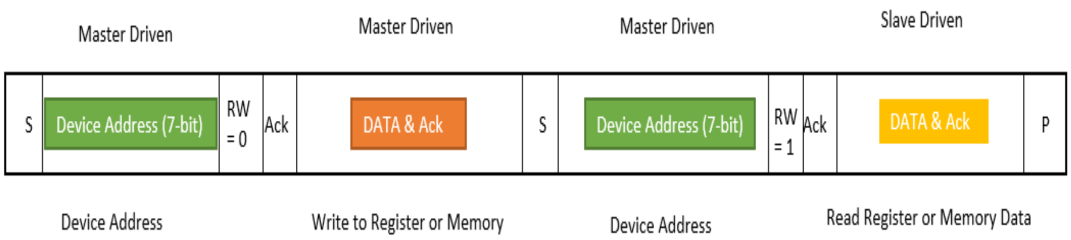

Fig. 3.1. I2C Write Read Diagram

In this work I2C bus was configured using simplicity studio `em_i2C` library. The I2C was enabled as a master with the default speed of 100 kHz. The same I2C port is used to connect sensors, as the sensor have different device addresses. The read and write cycles were performed to read and write the data respectively.

I2C `Write_Write`: The I2C `Write_Write` cycle is performed when we have to write data to the slave memory or register. Figure 3.2 shows I2C write write diagram.

I2C `Write_Read`: The I2C `Write_Read` cycle is performed when we have to read data from slave memory or register. Figure 3.1 shows I2C write read diagram.

2. **Serial Peripheral Interface (SPI):** SPI is three wire communication. It is preferred over I2C when the data rate higher than 400 kHz is required. SPI also supports data rate higher than 10 MHzs. SPI bus contains three signals, MISO (Master in Slave out), MOSI (Master out Slave in), and CLK (Clock). The select lines are used to differentiate sensors connecting to same SPI bus. SPI uses separate wires for data and clock and hence called as synchronous communication.

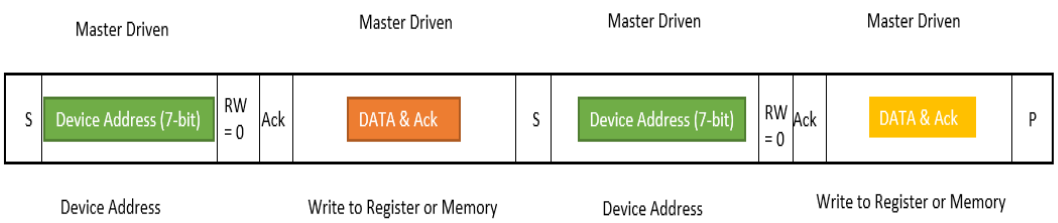

Fig. 3.2. I2C Write Write Diagram

In this work SPI is used to store data to SD card while data collection. The Simplicity studio MicroSD SPI library API is used to configure the SPI for SD card. The data is logged to SD card at default SPI speed.

3. **Universal Asynchronous Receiver Transmitter (UART):** UART is a piece of circuitry accountable for performing serial communication. UART is a mediator between serial and parallel. On one side of UART, a bus with eight data lines, and on the other side there are two serial wires with Rx and Tx. The transmitter and receiver need to operate on the same baud rate in order to communicate information on time. Along with the baud rate, the receiver and transmitter must have start bit, stop bit, data frame and parity. UART is used here to communicate with the BLE sensor. It is used to send AT commands at the baud rate of 9600.

3.2 Wireless Communication Protocol

Wireless communication can be short ranged or long ranged based on specific application and uses radio, sonic, electromagnetic and free-space optical as different modes to transfer information. The advancement in the field of wireless has added an important feature to embedded device development, circuit designing, home automation, automobile, and smart watches. The BLE protocol is used to establish the wireless communication in this work.

Bluetooth Low Energy: BLE or Bluetooth Low Energy is a short range protocol suitable for embedded applications. The low power in BLE is achieved by transmitting infrequent and small data packets with maximum bit rate of 30 KB/s. BLE is not suitable for high throughput streaming applications. BLE works on asynchronous connection, which means that the connection is terminated when data transfer is complete and the connection is made when the data is available. Connecting ⇒ transmitting ⇒ disconnecting ⇒ sleep. The operation of BLE can be explained by studying following modes.

- **Standby:** In standby mode, no device transmit or receives data.
- **Advertising:** In advertising mode the data is been broadcasted to listening scanner. The data can be a request to form a connection. No data transfer is guaranteed in this mode. Advertising mode uses Generic Access Profile (GAP) layer.
- **Scanning:** In scanning mode the device listens to advertising mode. It checks if any device wants to form a connection.
- **Initiating:** In Initiation state, the connection is established.
- **Connection:** In connection mode, the connected devices transmits server to client data. It is a one to one transfer with data guaranteed to be sent and verified. The connection mode uses generic attribute profile layer (GATT). GATT defines transmit and receive of short pieces of data called attributes between server and client.

To achieve low power and reliable performance, the BLE protocol is optimized from the classical Bluetooth. The optimization is done by sending short packets to reduce the transmit peak current and receive time, reducing RF channels to improve discovery and connection time, single protocol etc. Figure 3.3 shows the BLE protocol layers.

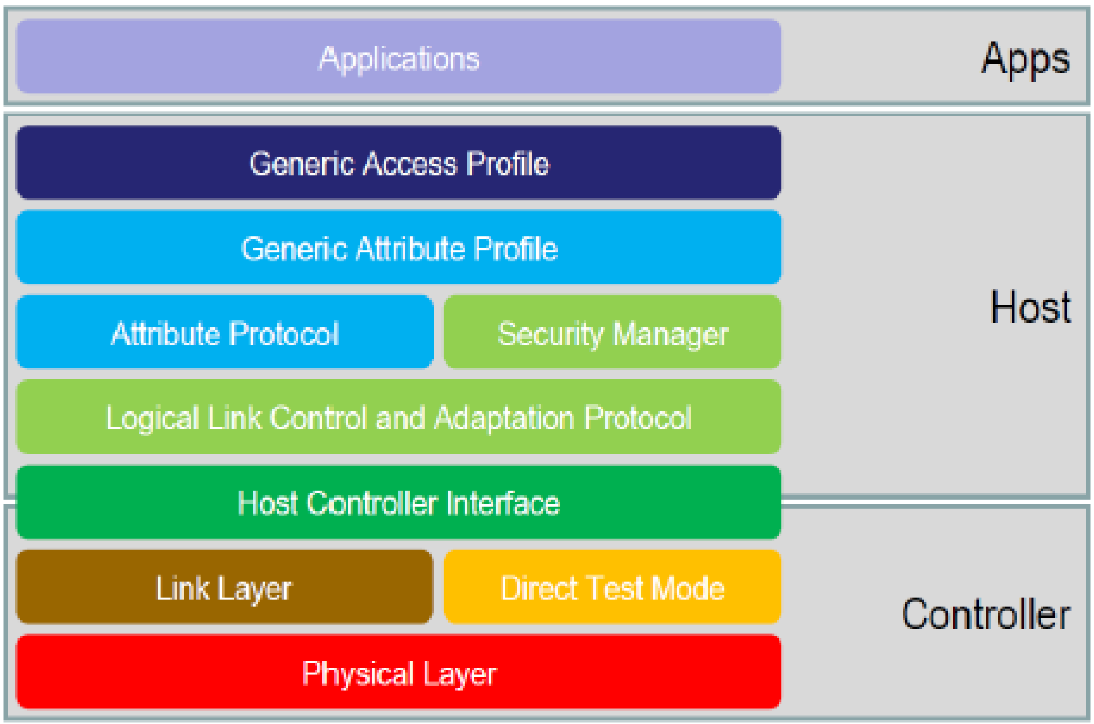

Fig. 3.3. The BLE Protocol Layers

4. SYSTEM ARCHITECTURE SOFTWARE DESIGN

The system architecture is the integrated software/hardware system that is designed to provide the minimum power consumption earliest possible fall detection. The power consumption is important for the wearable devices so the portable battery may last longer. The approach followed here is based on the system approach and the selection of the processor, the sensor, and the wearable devices.

4.1 The Block Diagram

Figure 4.1 gives the overview of low power Pre-Fall detection system designed in this work. The system is divided into the sensing unit, the controller unit and the communication unit. The system illustrates the relationship between these three units for better explanation of the system function.

Sensing unit: The sensing unit consists of the activity monitoring sensors namely accelerometer and gyroscope. This unit is responsible for reliable data acquisition from ADXL345 and MPU6050 sensors mounted on the human body as a wearable technology.

Controlling unit: The controlling unit is the heart of the system. It consists of EFM32GG microcontroller and HM-10 BLE sensor. Its functionality includes initializing motion sensors, calibrating motion sensors, reading raw data, processing and computing raw data, analyzing computed data, and deciding about the fall. The unit also helps in transmitting the fall information to the smart phone connected via bluetooth low energy.

Communication unit: The communication unit helps with transmitting the fall critical information to the caretaker, hospital, or health-care provider. This unit consists of the mobile application, which connects the bluetooth low energy to the

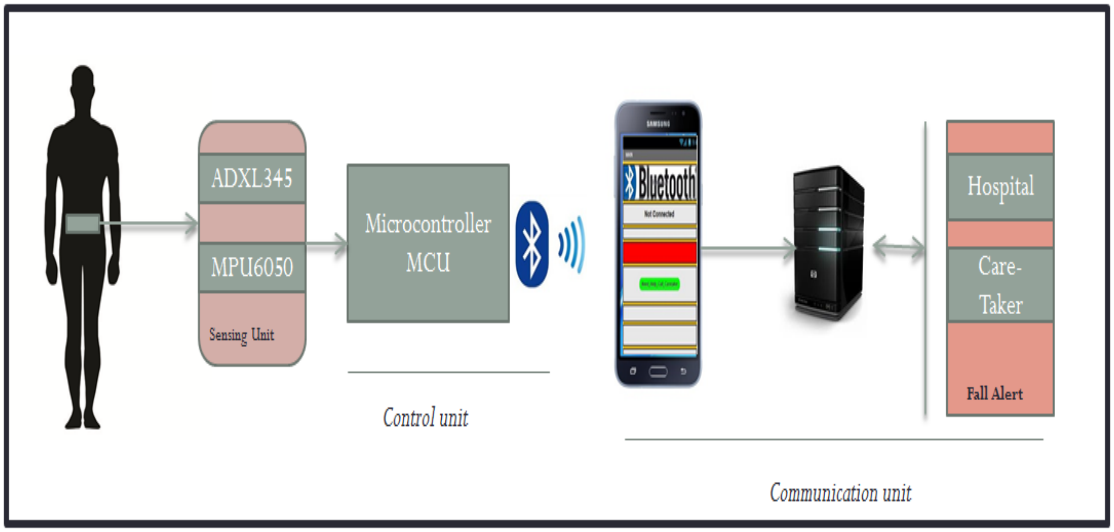

Fig. 4.1. The System Overview

one on the controlling unit. On fall, the communication unit receives the information and transmits it to the caretaker in the form of a message with the time of fall and GPS location.

4.2 Functional Diagram

The functional diagram in figure 4.2 shows the hardware connection of Pre-Fall detection system. The motion sensors, ADXL345 and MPU6050 communicates with control unit (microcontroller) using I2C protocol at the default speed of 100 kHz. The ADXL345 sensor is configured at full range of 13 bit ADC to sample data at 100Hz low power mode with the measurement range of ± 4g. The sensor is also configured to generate free-fall and activity interrupts by setting bits in register `INT_ENABLE`. This interrupts are mapped to INT1 and INT2 pins of ADXL345 by setting bits in register `INT_ENABLE`. The MPU6050 sensor uses internal clock source and 16 bit ADC to sample the data. It is configured to sample accelerometer and gyroscope data at 1

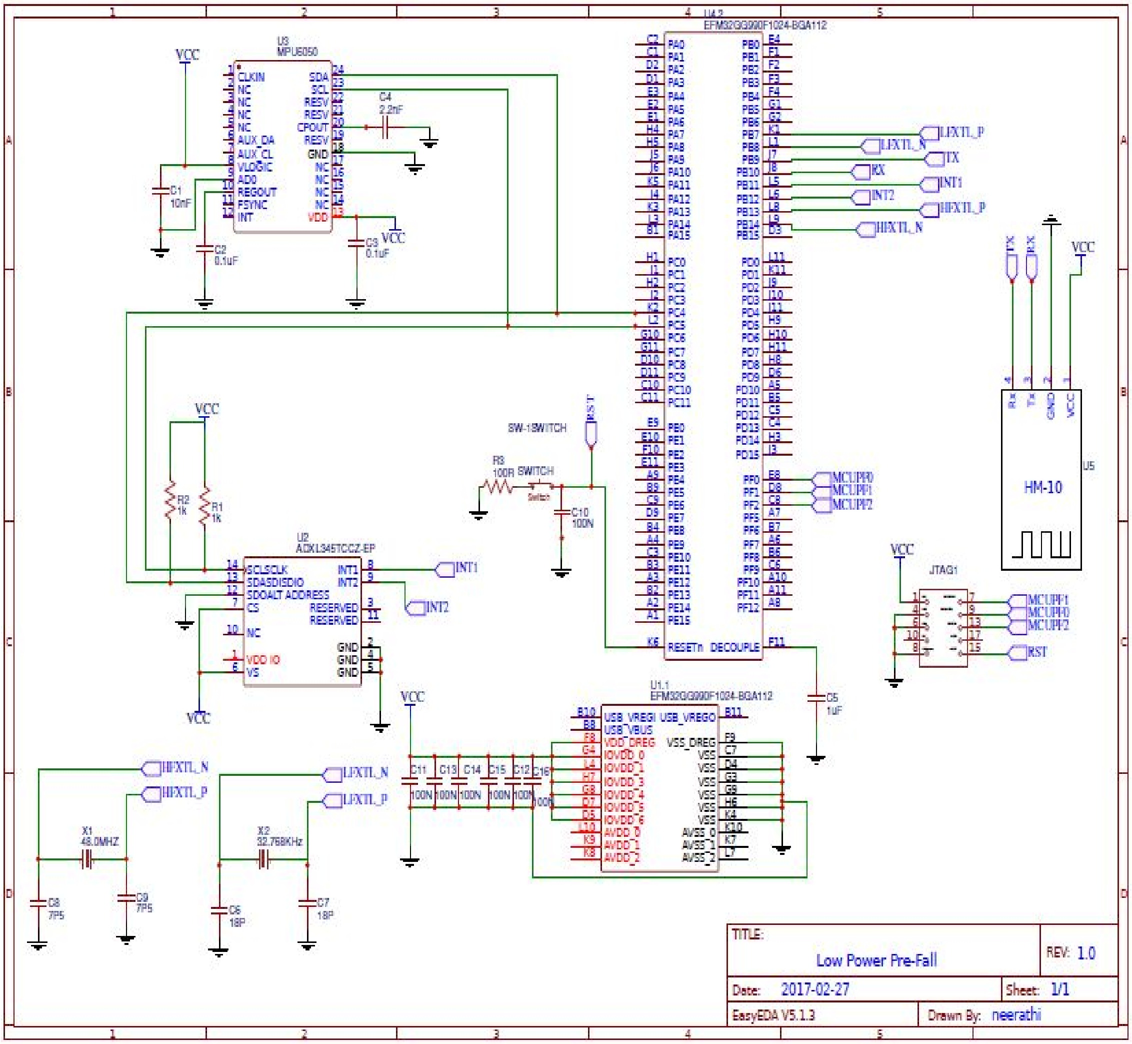

Fig. 4.2. The Hardware Connection Diagram

kHz with the accelerometer measurement range of ± 4g and gyroscope measurement range of 250 degree/s. The HM-10 communicates with control unit using UART at default baud rate of 9600. Tables 4.1, 4.2, and 4.3 give the pin-to-pin connection overview of the hardware design.

Table 4.1.
Connection between MCU to ADXL345

MCU Pins	**ADXL345 PINS**
I2C1 LOC0 PC4 (SDA)	SDA
I2C1 LOC0 PC5 (SCL)	SCL
GPIO PB11	INT1
GPIO PB12	INT2
VCC	VCC
VCC	CS
GND	GND

Table 4.2.
Connection between MCU to MPU6050

MCU Pins	**MPU6050 PINS**
I2C1 LOC0 PC4 (SDA)	SDA
I2C1 LOC0 PC5 (SCL)	SCL
VCC	VCC
VCC	VIO
GND	GND

Table 4.3.
Connection between MCU to HM-10

MCU Pins	**HM-10 PINS**
UART1 LOC2 PB9 (TX)	RX
UART1 LOC2 PB10 (RX)	TX
VCC	VCC
GND	GND

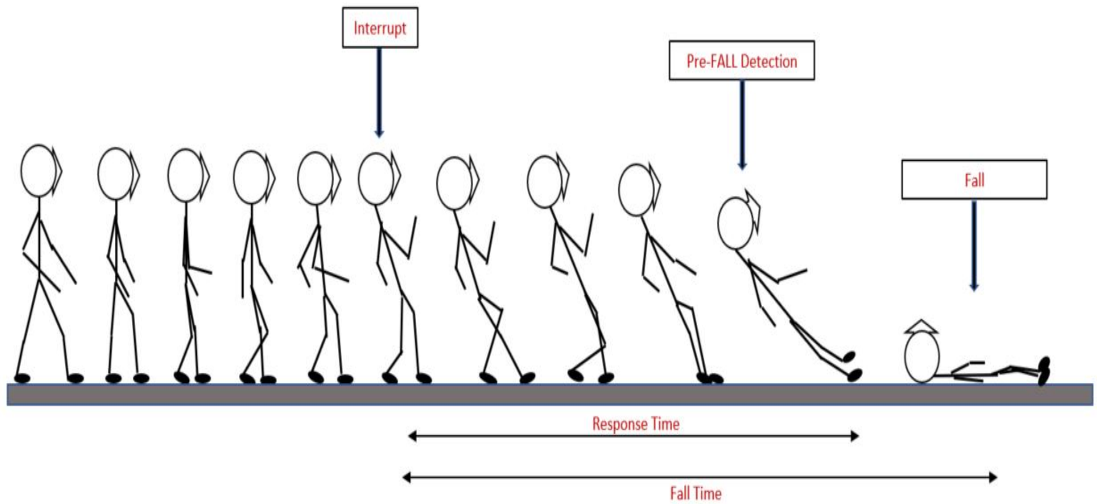

Fig. 4.3. The Pre-Fall Detection Approach

4.3 Software Design

The figure 4.3 shows the methodology of Pre-Fall detection. The approach is to sense high peaks in patient movements, which are caused by unstable actions, start of fall, and sudden sit. The high peak above the threshold generates an interrupt, waking up the MCU and motion sensor to track human body posture for next two seconds. The window of two seconds is used to classify patient's activity after the peak. It also helps in minimizing unrecognized true-fall events. To ease our understanding on fall dynamics, the following events are defined.

- **Interrupt Threshold:** Motion thresholds are the acceleration peaks generated during a sudden sit, shock moments, freefalls, and rest. During occurrence of fall there is either drop down in acceleration or sudden peak. Detecting it gives a starting point of sudden activity or fall.

- **Fall Time:** Fall time is the time when the interrupt is generated to the time when the fall happened.

- **Response Time:** Response time is the time taken by the Pre-Fall detection system to identify and declare fall since interrupt occurred.
- **Fall Detection:** Detecting a fall at a posture from where the person cannot return to the normal or balance state. The fall detection decision in this work is truly based on thresholds.

$$PrefallTime = FallTime - ResponseTime \tag{4.1}$$

The equation 4.1 gives us time at which the Pre-Fall detection system have detected a fall. This time helps us in designing a safety trigging system to protect patient.

4.4 The Flow Diagram

The Algorithm flow diagram is classified into two parts:

1. Initialization and deep sleep mode.
2. Interrupts and Interrupt service routine.

Figure 4.4 describes the flow chart of the initialization and deep sleep mode software flow. On reset or power down, the code starts from the start address. The first step on start is to initialize the Micro-controller (MCU). The initialization of the MCU consists of restoring the static RAM, initializing the peripherals (GPIO, I2C, and UART), configuring clock (Main and Peripheral) and disabling the unwanted peripherals and clock. After MCU initialization, the motion sensor checks for its connection and communication status. Both the motion sensors are connected on the same I2C bus/port. This is possible because both sensors have different device addresses. Further, we perform I2C WR (Write read) for ADXL345 and MPU6050 to read the device address. If the device address returns by the slave (ADXL and MPU6050), and it matches the expected data stored in the micro, we move forward, otherwise the fault LED lights up.

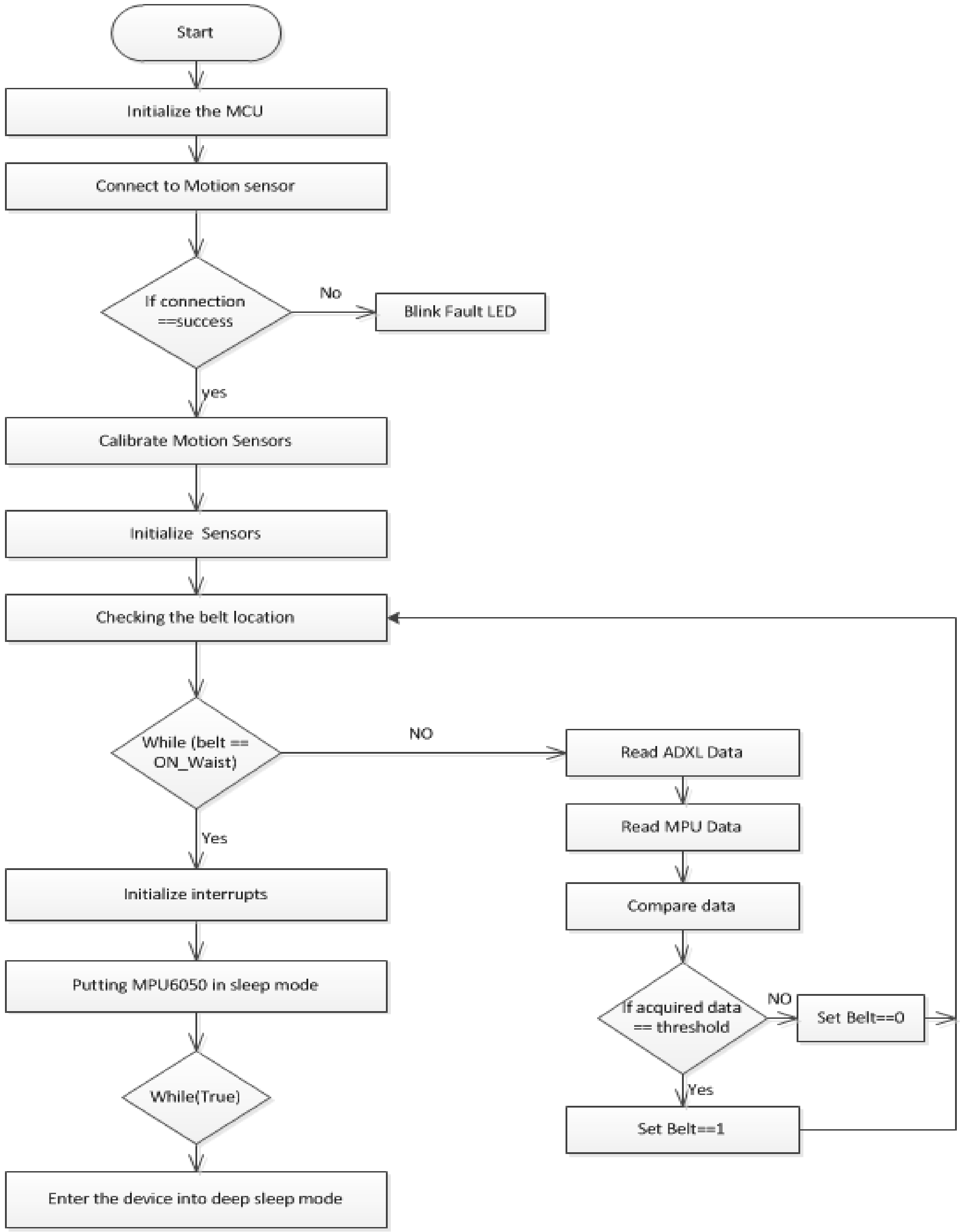

Fig. 4.4. The Flow Diagram to Initialize and Sleep

Once the sensor connection check is successful; we calibrate both the motion sensors. Sensor calibration is done by setting up an appropriate sensor register to configure the device, and get average count biases. Calibration is necessary because there are variations between the sensor data. For critical applications where precise measurements are required, calibration to the reference gives the desired result. After sensor calibration, we initialize the sensor to the desired configuration: setting up interrupt registers, data rate, resolution, range, low power mode etc. Now, the next step is to check the belt location. To detect the Pre-Fall accurately, it is of high importance to place the belt at the right location (around the waist). This step will keep on looping until the set thresholds are met. Once the actual raw values are in range, we break the loop and initialize the interrupts. This is done by setting MCU to expect the interrupt on specific pins with a callback. After we complete this setup successfully, we put the MPU6050 to sleeping mode. MPU6050 sensor data is only needed when calculating the fall parameters. Initially, it's just ADXL345 that operates at the low power mode to track users linear velocity. It generates an interrupt when motion above the threshold is achieved. Until the time we get interrupted by ADXL345 we put the processor in deep sleep mode. In deep sleep mode, the current consumption by the core is pulled down to 90μA at 3V supply voltage. Therefore, putting the MCU in deep sleep mode assists with the minimum power consumption. Figure 4.5 explains the Interrupts and Interrupt service routine flow in Pre-Fall detection algorithm. The interrupt from ADXL345 puts the device into RUN mode from Deep sleep mode. On interrupt, the device saves the current executing instruction address into the stack and jumps onto the interrupt service routine (ISR). Once we enter into ISR, we first disable the interrupt and enable MPU6050 from sleep mode. Disabling an interrupt is necessary to avoid MCU generating interrupts when we are calculating fall parameters.

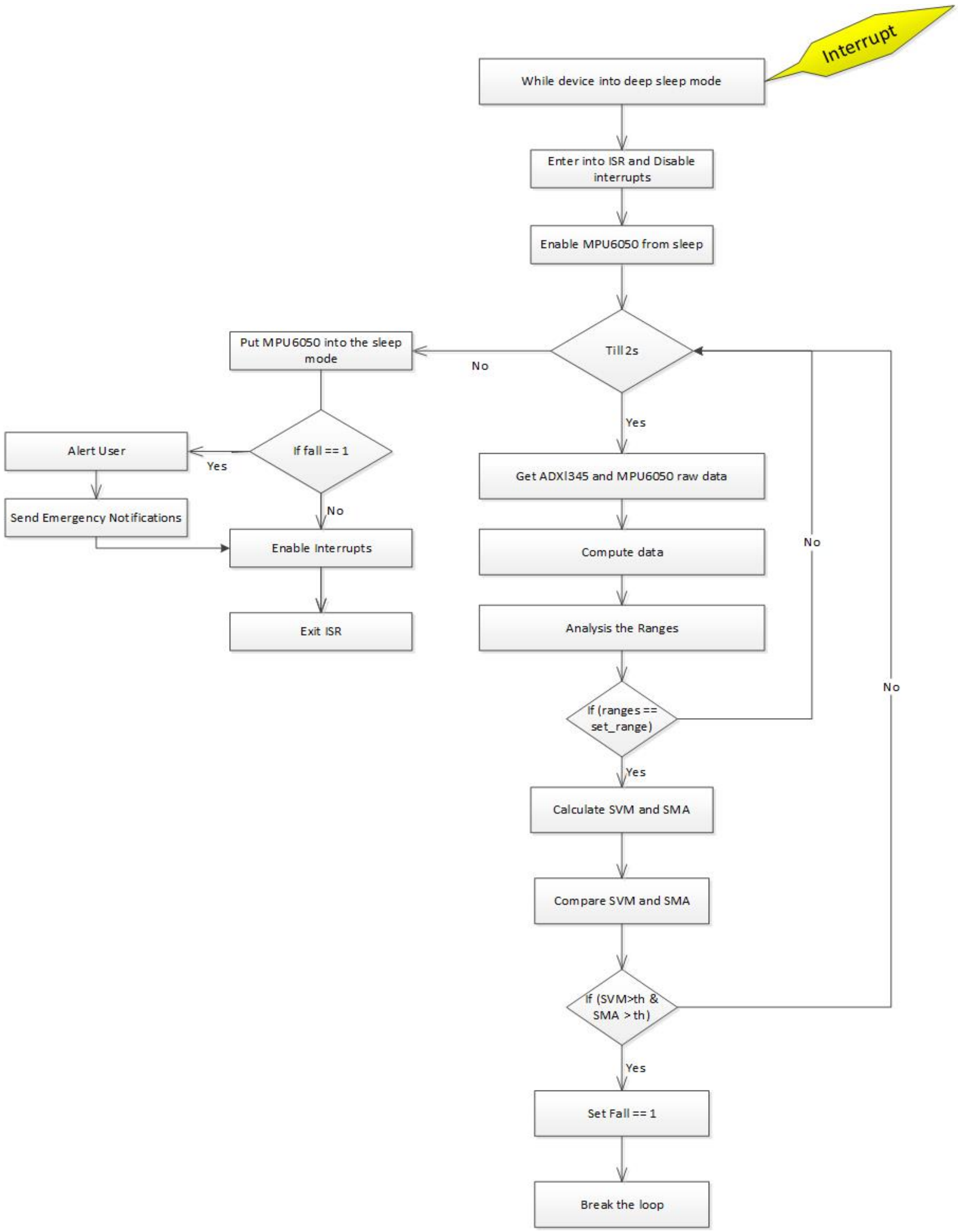

Fig. 4.5. The Flow Diagram to Handle Interrupts and Detect Pre-Fall

A loop of two seconds is used to track user activity after interrupt is detected. In the loop, we first get the raw data from MPU6050 and ADXL354. The raw data is then computed to remove drift and noise by using low pass filter for ADXL345 and complementary filter for MPU6050. The computed values are now angles, which are used to analyze users body posture.

Sudden sit, Bending, and few normal-inclining users body poster results in a range of fall. Therefore, an additional comparison is needed to successfully differentiate between the fall and activity of daily living (ADL). Once the user's body posture is classified to be in range, then we calculate and compare SVM and SMA with the stored thresholds. If both the values are greater than the set threshold, fall variable is set and we exit the loop. If the range condition and the SVM and SMA do not meet the threshold, we exit the loop, without updating the fall variable. After exiting the loop we first put the MPU6050 to sleep mode and then checks if fall variable is set. If the fall variable is set, we alert the user by sending emergency notification. The interrupts are then enabled and we exit the ISR to get into deep sleep mode.

4.5 Software Environment

In this project, a simplicity studio from silicon labs is used as development tool to build the software. It is a launching pad for anything needed to configure and develop with EFM32 MCU. It helps greatly with reducing the time and complexity by providing a high-powered IDE, hardware configuration tool, power consumption measurement tool, and links to helpful resources. As an Integrated Development Environment (IDE), simplicity studio also comes with an inbuilt compiler and debugger.

1. **Compiler:** The GNU GCC compiler is used to compile the software. It helps in optimizing the code and converting it to a machine language (HEX). The compiler built the code in regards to the hardware selected, because hardware architecture: ports, clock, memory, and peripherals plays crucial role in successfully generating the processor specific machine code.

2. **Debugger:** Segger J-link debugger is used to flash and debug the code. The debugger helps in identifying bugs by providing a way to pause/step the program and set break points to analysis the registers, variables, and functions.

3. **Energy Profiler:** The starter kit used includes Advanced Energy Monitoring (AEM). It is the hardware component installed on the kit to measure the current consumption. The samples of current consumption from AEM are sent over j-link debug to the IDE. AEM with the sampling frequency of 6250Hz measures current from 0.1μA to 50 mA with absolute accuracy of 1μA and relative accuracy of 0.1μA. The Energy profiler displays the AEM current sample waveform.

5. COMPUTATIONAL MODEL

In Pre-Fall detection processes, *accelerometers* ′ and *gyroscopes* ′ information are used to detect human fall. The implementation of mathematical model is designed to be less complex for low power computation. The reduced complexity and computation power helps to significantly reduce battery power consumption, while maintaining the fall detection accuracy and system performance.

In this approach, we detect the Pre-Fall by thresholding the velocity and acceleration in horizontal and vertical directions of users moment. In [37], authors described a single tri-axial accelerometer, which is enough for human fall detection as sufficient information to classify human body posture, and this information can be extracted from its measurements. The gyroscope is added to further reduce the false fall probability. Using both gyroscope and accelerometer helps to determine accurate static and dynamic human postures, at the same time it reduces computational cost and enhance the system accuracy [38].

5.1 The Accelerometer

Accelerometer measures acceleration, which is the rate of change of velocity of a subject. Acceleration can be measured in three directions (X, Y, and Z) simultaneously in G-forces (g). Accelerometer works by sensing either static (gravity) or dynamic (vibrations and movements) forces in acceleration. As acceleration holds direction and magnitude, it can be used to sense the orientation of the device. Accelerometer measures 1g when aligned with earth's downward gravitational field. So when the subject is tilted, the 1g get's distributed among the three axis. To determine the fall movement, the subject orientation can be measured using participation of two or three axis. Figure 5.1 shows the acceleration points in X, Y and Z axis with

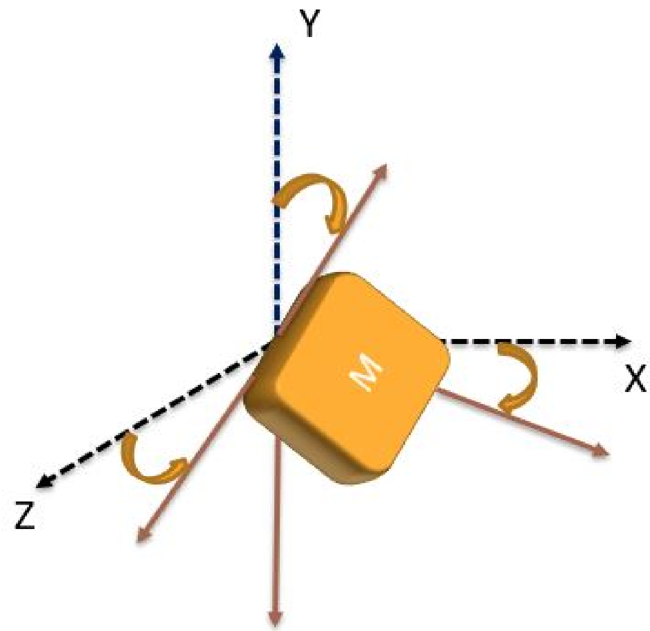

Fig. 5.1. Angle Measurement using Accelerometer

respective to acceleration vector. To have the best accuracy, all three axes are used to determine the angle. The three components of the acceleration vector are given as:

$$A_x = arctan(\frac{x}{\sqrt{y^2 + z^2}}) \tag{5.1}$$

$$A_y = arctan(\frac{y}{\sqrt{x^2 + z^2}}) \tag{5.2}$$

$$A_z = arctan(\frac{z}{\sqrt{x^2 + y^2}}) \tag{5.3}$$

During static conditions, the inclination of the accelerometer with respect to vertical is known. In Dynamic conditions, the inclination using accelerometer measurement is noisy since gravity is added to the system acceleration component. Digital filters were used to suppress the noise by separating the gravity vector and body acceleration vector.

5.2 The Gyroscope

Gyroscope measures angular velocities that are derived from the angular positions of gyroscopes. The rate of change of position over time, can be measured in degrees per second.

$$\theta_i = \frac{d\theta}{dt} \tag{5.4}$$

The angular position can be obtained by integrate the angular velocity. So, assuming that at t=0 we start with the initial angular velocity, $\theta_i(0)$. The angular position at any given moment t can be obtained from the following equation:

$$\theta = \int_0^t \theta_i(t)dt \approx \sum_{i=o}^{t} \theta_i(t)T_s \tag{5.5}$$

As seen from the above equation, the third part is the approximated value that we assume while using digital systems.

5.3 The Digital Filter

The most important and essential part for detecting the fall accurately is the measurements acquired from accelerometer and gyroscope. The raw values do not provide precise position and orientation of the object. Therefore, the digital filter is needed to get the optimized value of the signals measured from the sensors. Below are the problems associated with accelerometer and gyroscope raw output.

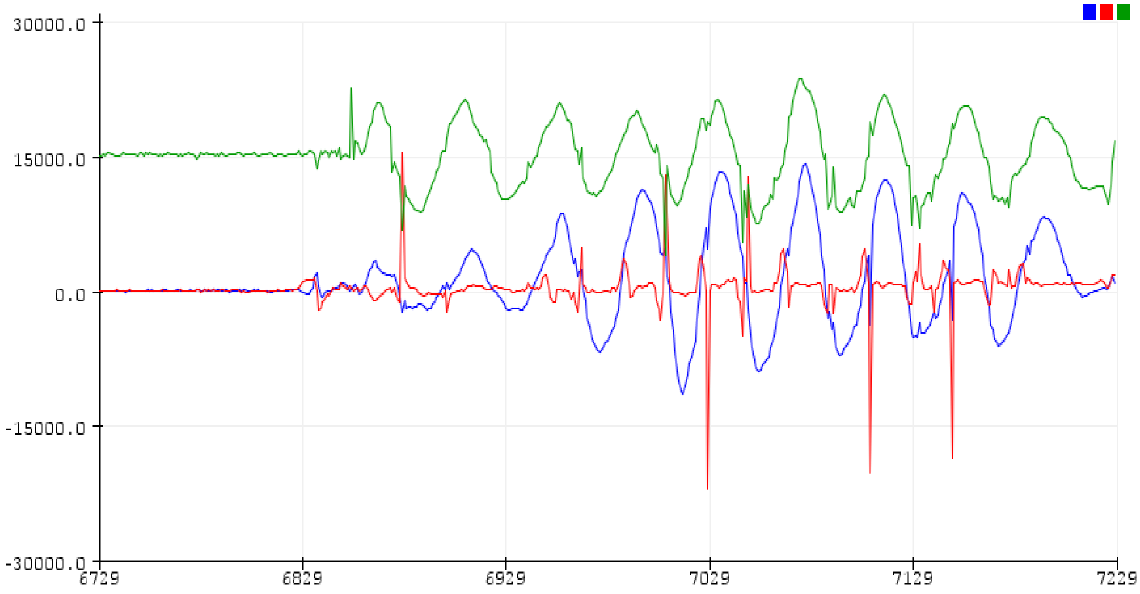

Fig. 5.2. The Accelerometer Raw Data

- **Noise in Accelerometer:** Accelerometer results in accurate rotation angles as long as the only force acting on it is gravity. When moving or rotating, the accelerometer value causes fluctuation in its reading, making it prone to noise. In Pre-Fall system, as user wear accelerometer, the body acceleration gets add up with the gravity acceleration, creating uncertainties in accelerometer measurement. Figure 5.2 shows the noise in accelerometer data.

- **Noise in Gyroscope:** The gyroscope measures angular velocity (ω) at interval (t), which is converted into angular position (ω*t) by integrating the gyroscope measurement data. The integration leads to repeatedly adding up computed intervals, generating bias error. This generates a gyroscope drift, when used for a longer time. Hence, it does not return back to zero when the system went back to its original position. Figure 5.3 shows the noise in the gyroscope data.

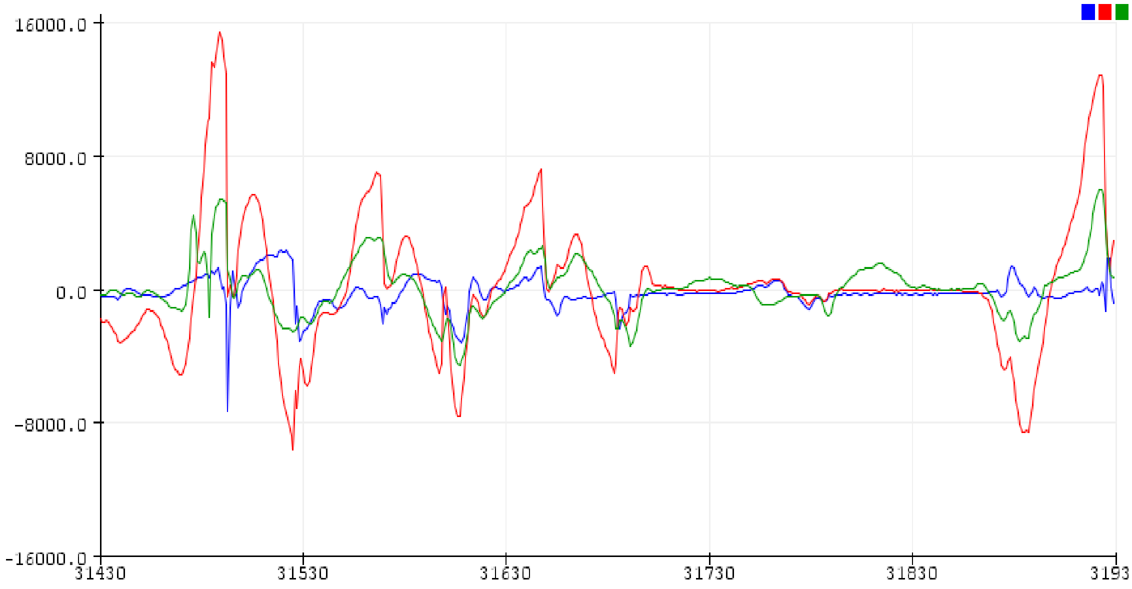

Fig. 5.3. The Gyroscope Raw Data

5.4 The Complementary Filter

The use of accelerometer and gyroscope for fall detection in real time situations, requires a math filter to combine the measured data returned by the sensors. A complementary filter is used where one source gives good information about low frequency signal while the other gives good information about high frequency signals. Complementary filter manages low pass as well as high pass filters simultaneously. The idea behind using complementary filter is to combine slow moving signals (long term signals) from accelerometer and fast moving signals (short term signals) from gyroscope to provide accurate information.

In static conditions, the accelerometer gives accurate values of orientation and in dynamic situations; the gyroscope gives the accurate value of tilts. Hence, the overall best solution for detecting fall, is to pass the accelerometer signals through a low pass filter and gyroscope signals through a high pass filter. Then combining the outputs (low pass and high pass), and multiplying with gains summing to unity to get filtered angle. The Complementary filter results in accurate data filtering with minimum complexity and computation requirements as compared to Kalman filter. Kalman filtering has been thought of for its widely used applications in aerospace, navigation systems and high precision applications. Some of the issues that discourages applying it in the low power consumption, was related to its system complexity, computational inefficiency, and CPU load. In particular, for complexity, it is complex to implement Kalman filter on top of the 32bit powerful ARM cortex M3 MCU. Because the filter needs to calculate the coefficients of the matrices, measurement error, and the process based error, etc are not trivial for real time application. In addition, the sensors output that can be used may result from Gaussian distribution. Above all, the use of kalman filter increases the execution time, hence increases the power consumption. Therefore, the complementary filter was used in this application.

The estimated θ_t is estimated via the expression:

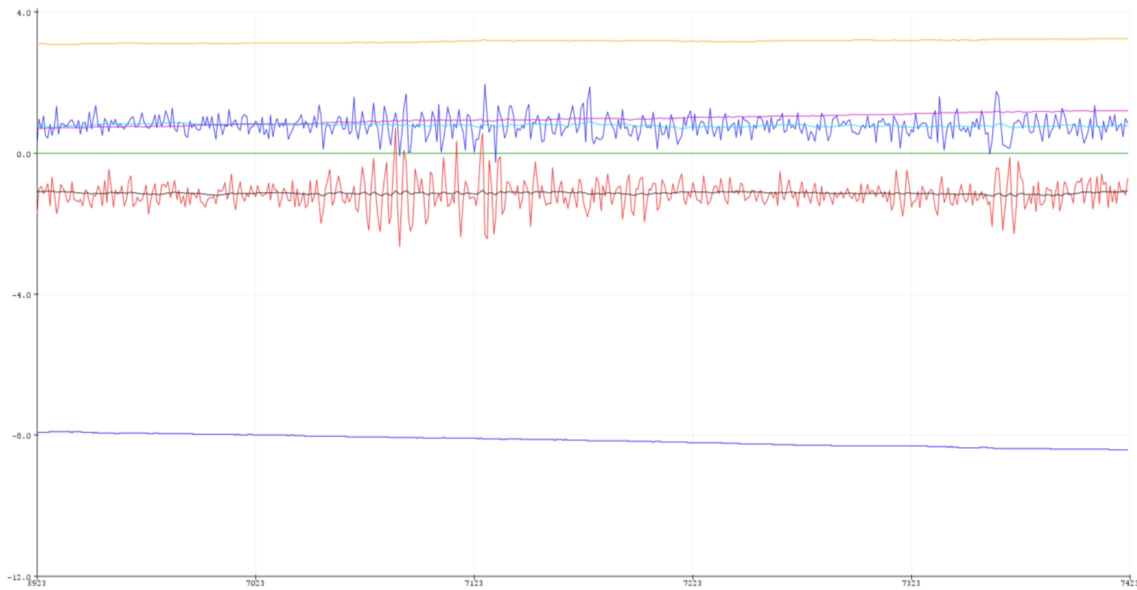

Fig. 5.4. Comparison of Complementary Filter Design with Raw Data

$$Comp_{angle} = (0.97) * (Comp_{angle} + Gyro_{data} * dt) + (0.03) * Accel_{data} \tag{5.6}$$

The complementary design is used to estimate both the Roll angle and Pitch angles. Equation (5.6) shows complementary filter formula. The gyroscope output is integrated with the current angle. Then it is combined with the accelerometer value, which is processed with equation (5.1). The constants 0.97 and 0.03 are filter gains. This indicates that the filter will take 97% of the gyroscope data, and added to the 2% of the accelerometer data. It can be changed to tune the filter based on the application. Figure 5.4 shows the results obtained after applying the complimentary filter to accelerometer and gyroscope output. The complementary filter removes noise from the accelerometer and eliminates gyro drift.

5.5 The Fall Parameters

5.5.1 Identifying Motion and Rest

It is necessary to distinguish between user activity and rest condition, with a suitable measure that include signal variations in all three axis. Such measure is described in [39], and is Signal Magnitude Area (SMA). SMA gives the static and dynamic positions of the patient by integrating the acceleration vectors (X, Y and Z) averaged over sampling period and summing it.

$$SMA = \frac{1}{t}(\int_0^t |x(t)|dt + \int_0^t |y(t)|dt + \int_0^t |z(t)|dt) \tag{5.7}$$

x(t), y(t), and z(t) refer to the body components of the x-, y-, and z-axis samples. Practically determining the acceleration values for motion (dynamic activity) and rest (static activity) sets an SVM threshold.

5.5.2 Postural Orientation Information

Postural information is required to identify relative tilt of body in space. To differentiate between ADL and fall it is important that we provide a distinction between them. Signal Vector Magnitude (SVM) is used to determine the fall. SVM provides a measure of movement intensity derived from accelerometer signal. The fall is determined by monitoring change in magnitude of acceleration, because it decreases rapidly during fall and before impact. During fall the body acceleration vector will be in the direction of gravity vector. According to [39] SVM_i is calculated at the i^{th} sample of the acceleration values on the X, Y and Z axis.

$$SVM_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \tag{5.8}$$

6. LOW POWER DESIGN AND ESTIMATION

The low power approach in this work focuses on minimizing energy consumption of microcontroller based systems. The low voltage CMOS process have significantly contributed in lowering the power consumed by the micro-controller, but there is still need of work to design a low power system than just fabrication process . Our aim is to select ultra-low power micro-controller and optimize the fall detection algorithm using the MCU features to extend battery life. The MCU selection was done by considering power consumption in low power and run modes, low power peripherals, and different operating frequencies. In this chapter, we focus on innovative approach taken to lower energy demands of wearable Pre-Fall detection system.

The MCU consumes most power since it handles most arithmetic and logic operations, among others. Therefore, it is important to understand the cause of power consumption and steps to minimize it. As discussed in chapter 1, the power consumption can be defined by Static and Dynamic power. The Dynamic power is defined as the power consumed when the micro-controller is running and performing its programmed tasks. While the static power is defined as, a power consumed when the micro-controller is ideal (not running the code).Therefore, to calculate the overall average power consumption it is important to know the time application expects to spend in deep-sleep and active.

- **Active Mode:** The Power consumption of a CPU, when active and processing the algorithm is highly important because of the higher power dissipation in active mode. The power loss in active mode happens due to dynamic power (6.1), which consume power while switching CMOS circuits. In (6.1), V is the supply voltage, f is the switching frequency and C is the load capacitance.

$$DynamicPower = V^2 * f * C \tag{6.1}$$

Therefore, by reducing the dynamic power, the total power consumption in active mode can be minimized. As the supply voltage (V) is a squared term in dynamic power equation, reducing it will significantly help in lowering dynamic power. The frequency of the application can also be lowered depending on the communication or sampling speed requirement, and processing performance. Therefore, from the (6.1), we can also say that lowering the operating frequency will result in lower dynamic power. The equation (6.3) gives the energy consumption in active mode.

(6.1) can also be represented as:

$$DynamicPower = V * (C * V * F)$$

$$DynamicPower = V * I \tag{6.2}$$

Where, I = Dynamic Current = (C * V * F)

From (6.2), Energy is defined as:

$$ActiveMode_{Energy} = Dynamic_{Power} * Time$$

$$= I_{Active} * Time_{Active} * V_{Active} \tag{6.3}$$

- **Deep sleep mode:** Pre-Fall algorithm is designed to spend the majority of time in low power deep-sleep mode waiting for external interrupt-events to wake-up the CPU. The power loses in deep sleep mode happens due to static power. While the code not actively running, the current consumption is due to bias currents for analog circuits, and leakage current. Static power consumption is important factor for pre-fall detection device, as the majority of time is spent in sleep mode. The static power can be reduced by using a MCU with designed advanced power management unit designed specifically to reduce leakage current. Other techniques used to reduce static power are disabling analog blocks, disabling ram blocks, and reducing supply voltage.

$$DeepSleepMode_{Energy} = I_{Deepsleep} * Time_{Deepsleep} * V_{Deepsleep} \tag{6.4}$$

Therefore, from above the total energy can be calculated as,

$$\begin{aligned} Total_{Energy} &= ActiveMode_{Energy} + DeepSleepMode_{Energy} \\ &= (I_{Deepsleep} * Time_{Deepsleep} * V_{Deepsleep}) + (I_{Active} * Time_{Active} * V_{Active}) \end{aligned} \tag{6.5}$$

Average Current Consumption can be represented as,

$$Average_{Current} = \frac{(I_{Deepsleep} * Time_{Deepsleep}) + (I_{Active} * Time_{Active})}{Time_{Deepsleep} + Time_{Active}} \tag{6.6}$$

Analyzing further, the power consumption is not restricted to deep sleep and active mode. Power is also consumed when there is transition for deep-sleep to active as show in Figure 6.1. During wake-up, a significant amount of power is used in preparing MCU to acquire data and process it.

Therefore, the total energy consumed is given by,

$$Total_{Energy} = ActiveMode_{Energy} + DeepSleepMode_{Energy} + Wakeup_{Energy}$$

The following steps are taken to achieve reduced current consumption:

1. **Using Energy Modes:** The most efficient way to save energy is spending less time in Active mode. when active, the MCU alone consumes 150 μA/MHz(2.3mA/14MHz). On the other hand, when CPU is in deep sleep mode it consumes 1.7μA. The API from `em_emu` library is used to switch between Active and deep sleep modes.

2. **Using Low energy peripherals:** The EFM32GG is built for ultra-low power capabilities. The availability of peripheral or modules varies as you move from one energy mode to another. This is done in order to save power. The MCU

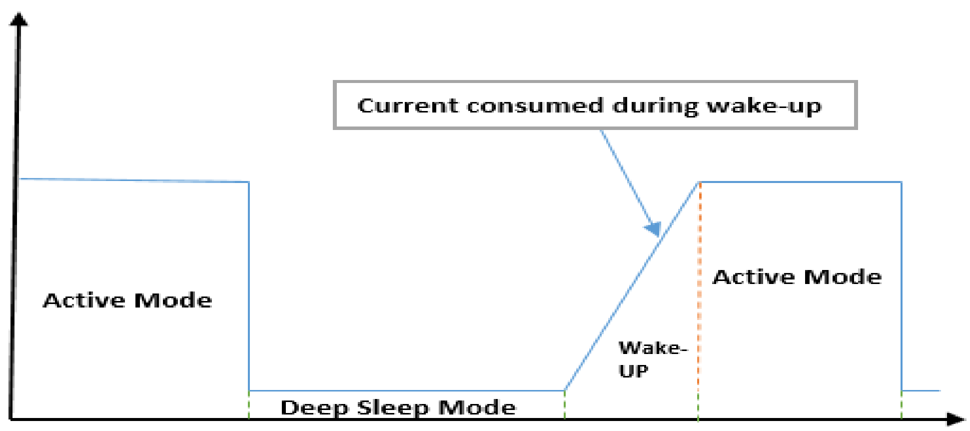

Fig. 6.1. The Current Consumption

comes with peripherals which are standard, but a layer of low energy is added on the top of it to make it low power. UART, I2C and SPI are redesigned to use lower energy in EFM32GG architecture.

- **LEUART:** Low energy UART is a basic UART with a layering of dedicated low power hardware on the top of it. LUART uses 9600 baud rate with current consumption of 150nA in sleep. The advantages of using LEUART is it receives and transmit data packets while CPU is in deep sleep mode.

3. **Disabling unused peripherals/modules:** All the unconnected GPIO and other peripherals are disabled to avoid GPIO leakage and save energy.

4. **Turning off clocks to unused peripherals/ modules:** The clock to unused peripherals contributes to energy consumption, even though the peripherals are disabled. The energy get still consumed by the module if the clock is still running. Therefore, it is energy efficient to turn off the clock for unused peripherals.

5. **Optimizing clock frequency:** As the energy consumed is directly proportional to the clock speed in CMOS circuit, it is important that the proper steps are taken to achieve the power management. In EFM32 several clock source and frequency choices are available to run the system at desired speed: HFRCO, HFXO, LFRCO and LFXO. The HFRCO and HFXO is a high frequency source clock. While HFXO uses external crystal resonator with 4-48MHz, the HFRCO uses internal RC with runtime selectable 1, 7, 11, 14, 21 or 28 MHz. The HFRCO clock was selected to run at 14 MHz. The pre-scaling of clock is avoided, because prescaler logic consumes addition power than that of using direct oscillator.

6. **Using lower operating voltage:** Lowering the voltage further reduces power consumption. Looking at the sensor specifications ADXL345, MPU6050, HM-10 and EFM32GG work at power of 3V.

7. **Optimizing Libraries:** The low level drivers developed for the MCU and sensors contains the APIs suitable for all applications. It is therefore important to remove the unwanted APIs to increase speed, reduce code size, and energy optimization.

7. RESULT AND DISCUSSION

The test of the fall detection system has been conducted based on the hardware and software structure described above. The prototype was designed and tested in the lab environment. The data was analyzed from the test results gained from 3 females and 4 males subject's performing simulated falls and activity of daily living. The subject's age, body mass and height lie between 25-40 years, 45-80 kg, and 167-182 cm respectively. Figure 7.1 shows the designed system for Pre-Fall detection.

Fig. 7.1. The Practical Model

7.1 Data Collection System

The data collection system was developed to:

- Analyze the fall data: To computed complimentary filter data, low-pass filter data, and SVM and SMA data.
- Examine the thresholds for activity of daily living and fall.
- Handle false triggered fall.

Data collection is necessary to evaluate how user's daily activities are; what acceleration range they belong into; what action triggers peak values; finding fall pattern; finalizing the thresholds; and examine motion sensor position. Figure 7.2 shows the data collection system design and Figure 7.3 shows the software flow diagram of data logging.

7.2 Bluetooth Mobile application

The Bluetooth mobile application was develop to receive fall signals from the detection system and send emergency message to caretaker. The Application utilizes the smart phone built in BLE, GSM and location sensor to receive and send message with GPS coordinates. It supports panic button to call and share location with care taker. The application also notifies subject that detection unit has detected the fall and an emergency text has been sent to caretaker. The application was developed in MIT app inventor. It supports and builds application for android-based smart phones. The app builder was used because of its simplicity and reliable programming structure. The Figure 7.4 shows mobile application. The Figure 7.5 shows results.

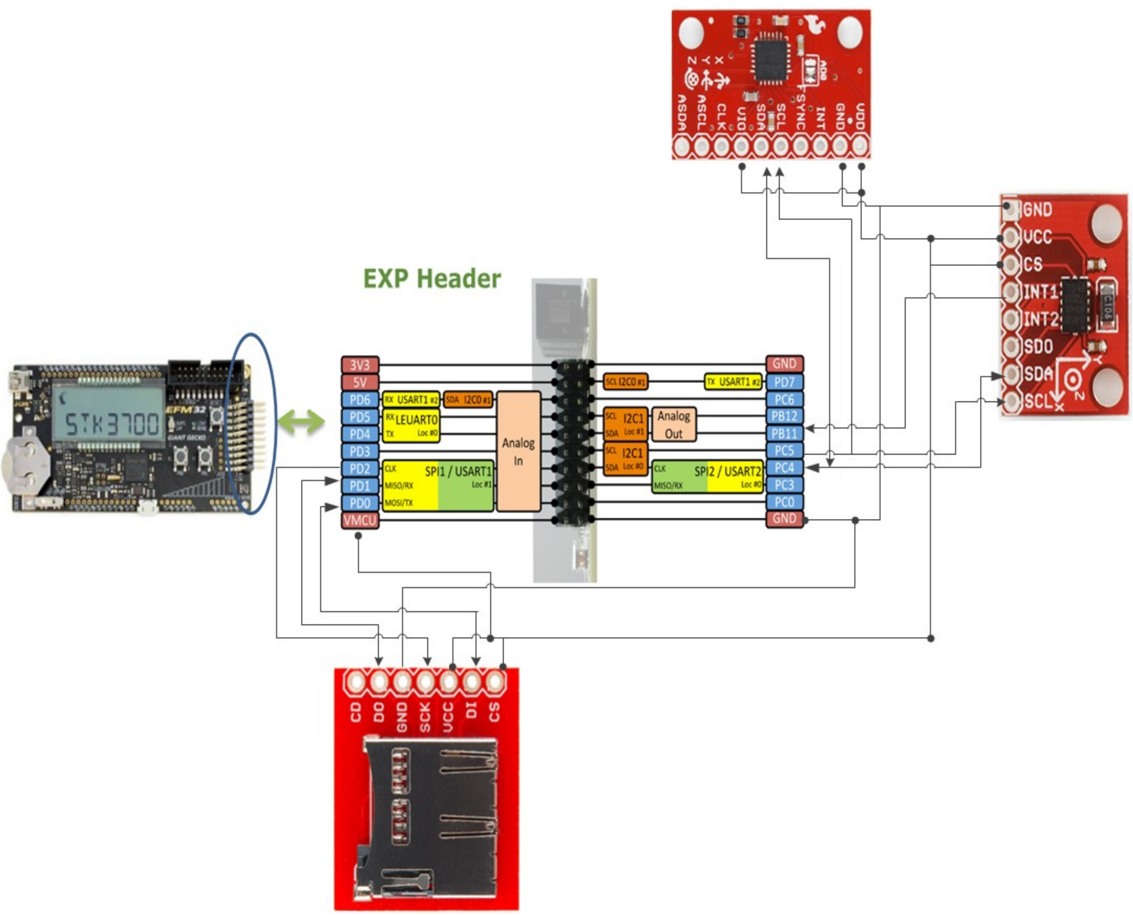

Fig. 7.2. The Data Logging System Design

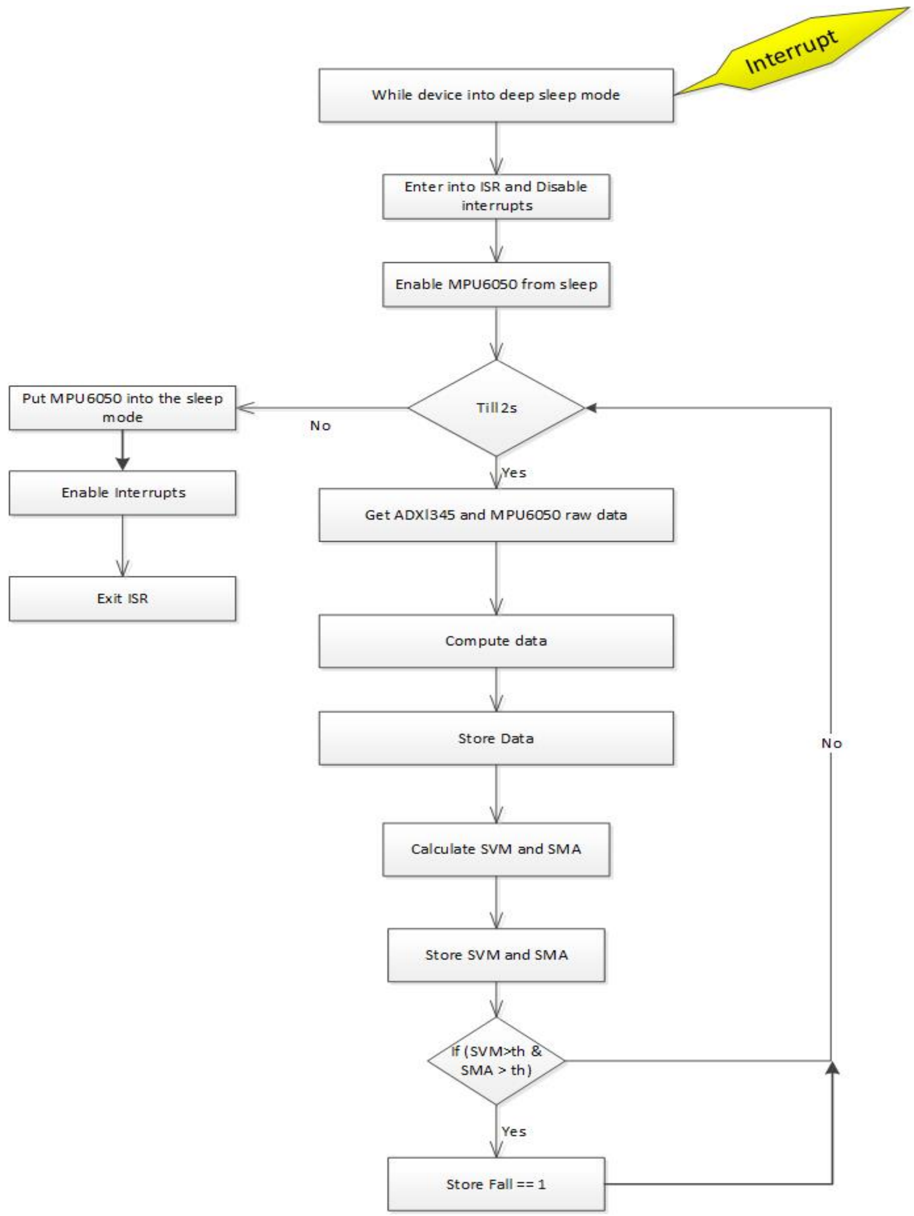

Fig. 7.3. The Data Logging Flow Diagram

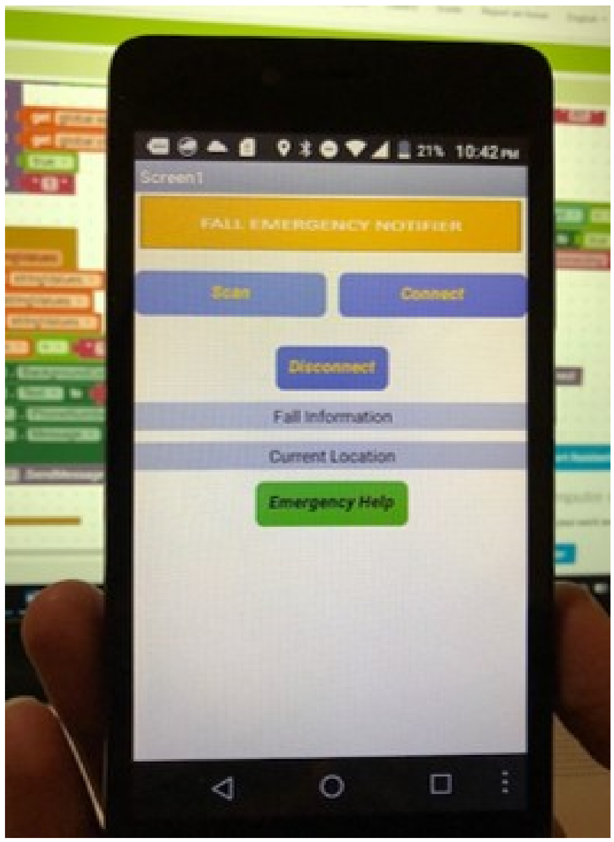

Fig. 7.4. The Mobile Application Screen

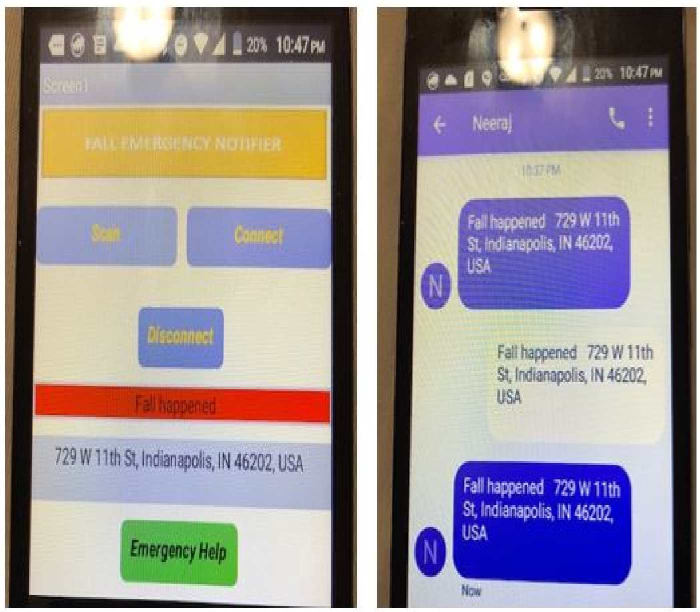

Fig. 7.5. Fall Result Using BLE App

7.3 Low power Test Results

The power consumption of the system: Microcontroller, MPU6050, ADXL345 and HM-10 is been monitored through Silicon labs Energy profiler. The current consumption was brought down to $\approx 600\mu$A when processor in sleep mode and $\approx$ 6mA during run mode. The current consumption gets higher when fall happens, because BLE sensor wakes up from sleep to send data.

1. **MCU Power Consumption:** Figure 7.6 shows the power consumption when MCU is in deep sleep mode and no sensors are connected to the system. The average current consumed by microcontroller in deep sleep mode is 1.7 μA. Figure 7.7 shows the power consumption when MCU is running in Active mode at 14MHz. The average current consumption in active mode is 2.3mA.

2. **ADXL345 Power Consumption:** Figure 7.8 shows the power consumption of an accelerometer and MCU. The red line shows the applied voltage to the sensor and MCU. The ADXL345 is configured to operate in low power mode. It measures the acceleration data while MCU in sleep mode to monitor subjects body posture. The power consumed by the ADXl345 and MCU during deep sleep mode operation is 83.66μA. According to data sheet at the low power data rate of 100Hz, the sensor must consume 55μA, when supplied 2.5V. In our case the supply voltage is 3.3 V therefore the current consumed is slightly higher.

3. **MPU6050 Power Consumption:** Moving forward we configure MPU6050 to operate in low power mode. Figure 7.9 shows MCU and MPU6050 current consumption when MCU acting in deep sleep mode and MPU6050 in measurement mode. As it can be seen the current consume by MPU6050 is 3.80mA.To overcome this problem, we put mpu6050 to sleep mode. Figure 7.10 show the power consumed by MPU6050 when in sleep mode. The current consumption for MPU6050 and MCU is now 8.57μA.

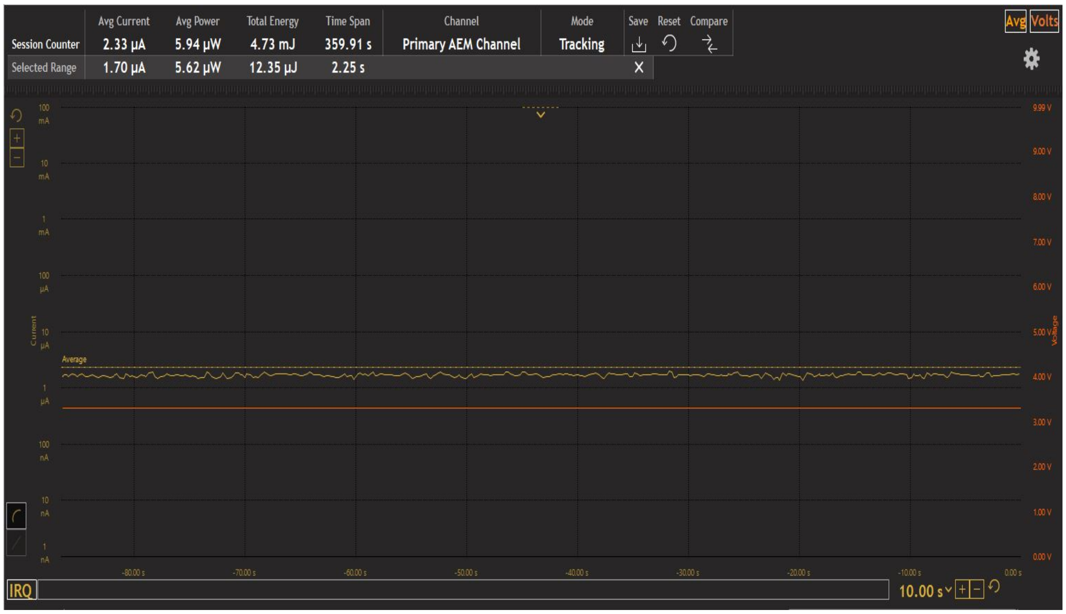

Fig. 7.6. The MCU Deep Sleep Mode Power Consumption

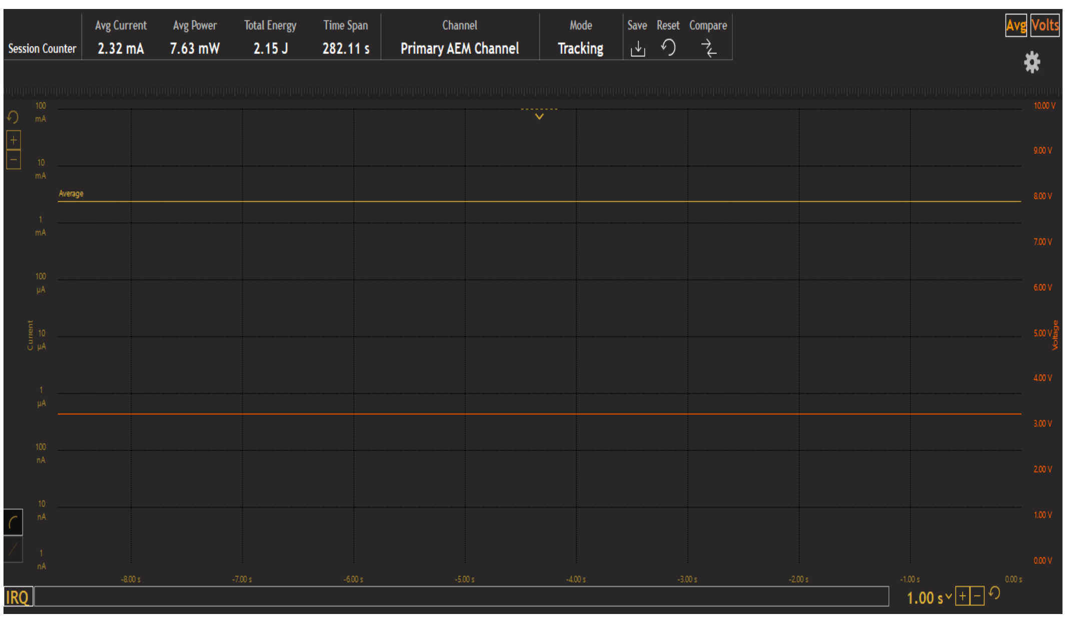

Fig. 7.7. The MCU Active Power Consumption

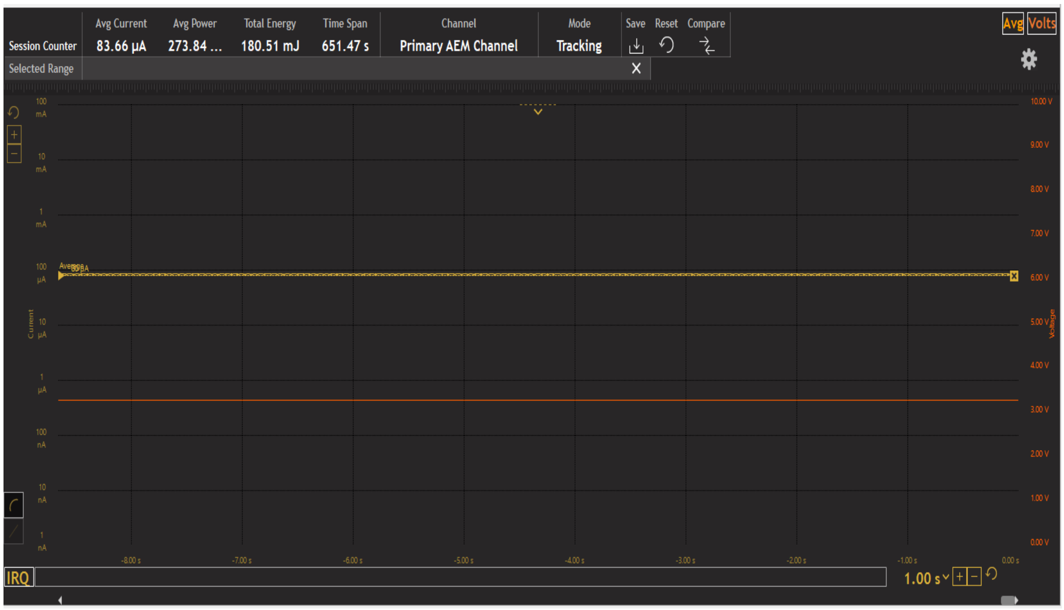

Fig. 7.8. The ADXL345 and MCU Deep Sleep Mode Power Consumption

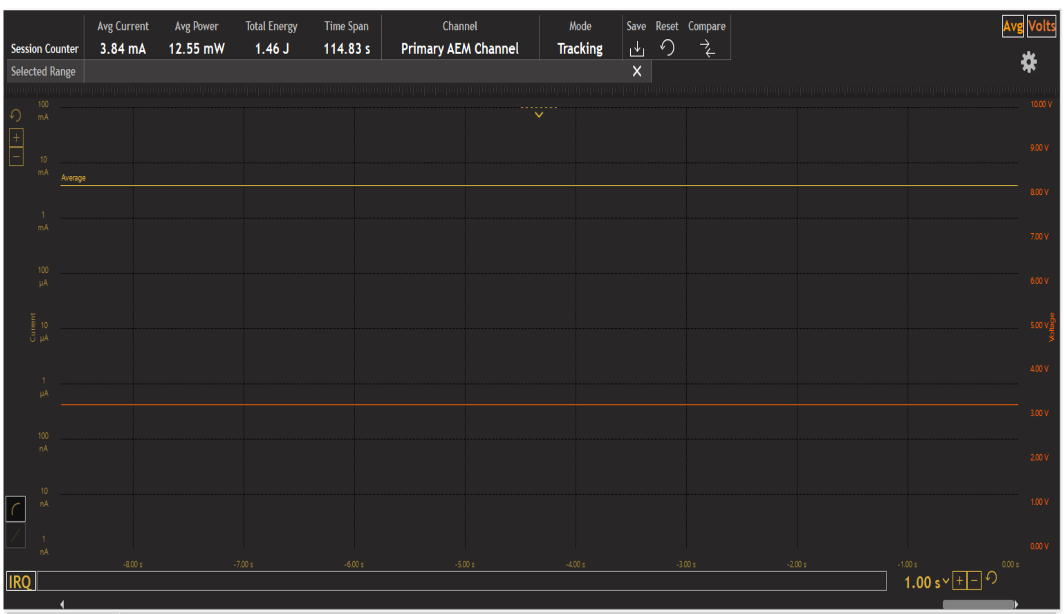

Fig. 7.9. The MPU6050 Active Mode Power Consumption

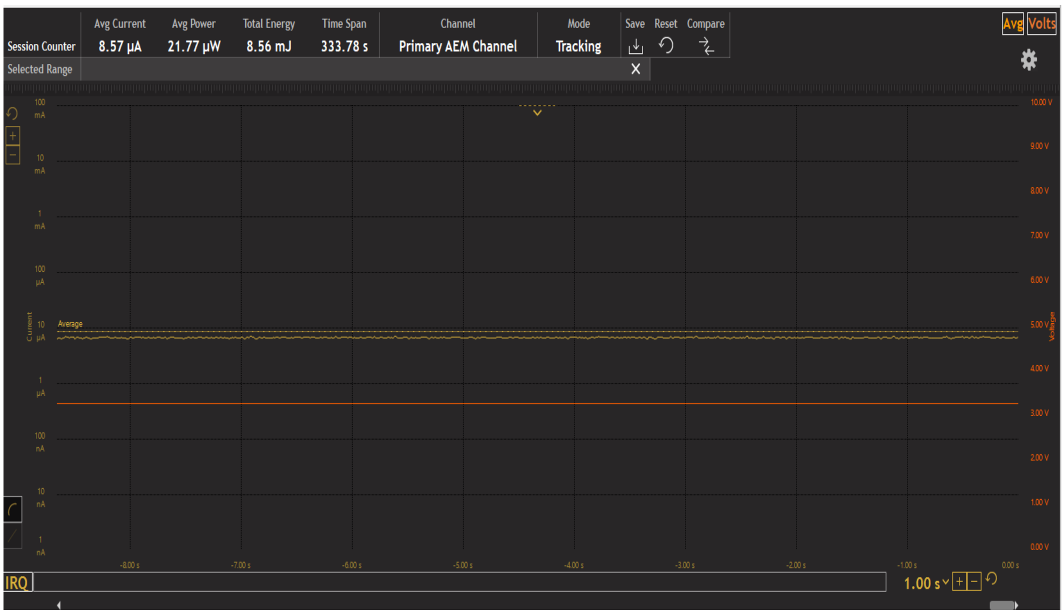

Fig. 7.10. The MPU6050 Sleep Mode Power Consumption

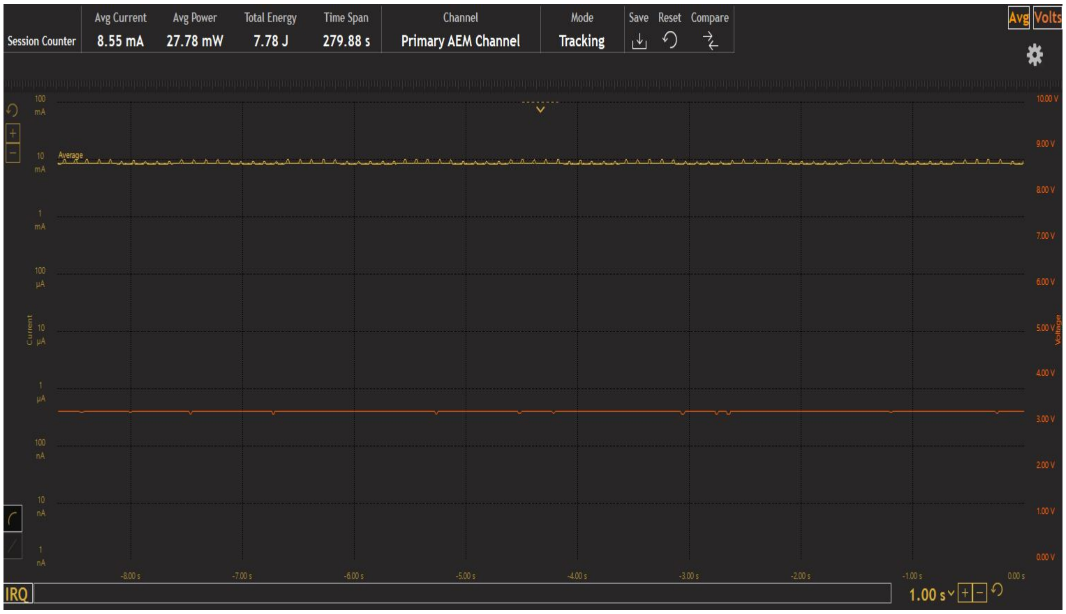

Fig. 7.11. The BLE Connected Mode Power Consumption

4. **BLE Power Consumption:** The Power consumed by HM10 in connected and Sleep mode is shown in figure 7.11 and figure 7.12. The module consumes approx. 8.55 mA in connected mode and 526.55μA in sleep mode.

The figure 7.13 shows the total current consumed by MPU6050, ADXL345 and MCU in active and sleep mode. In low power mode ADXL345 consumes 78μA and with MPU6050 and MCU it adds ups to 90μA. When interrupt occurs, we wake up MPU6050 which consumes 3.8mA. The ADXL345 power consumption stays the same 78μA and the 2.3mA is consumed by MCU to make a total of 6.1mA.

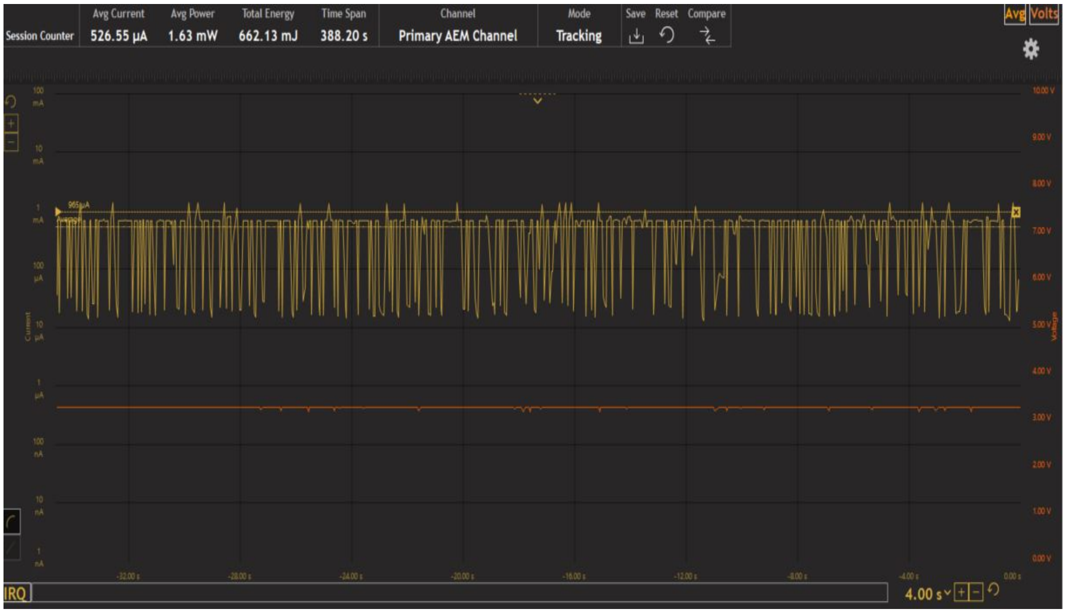

Fig. 7.12. The BLE Sleep Mode Power Consumption

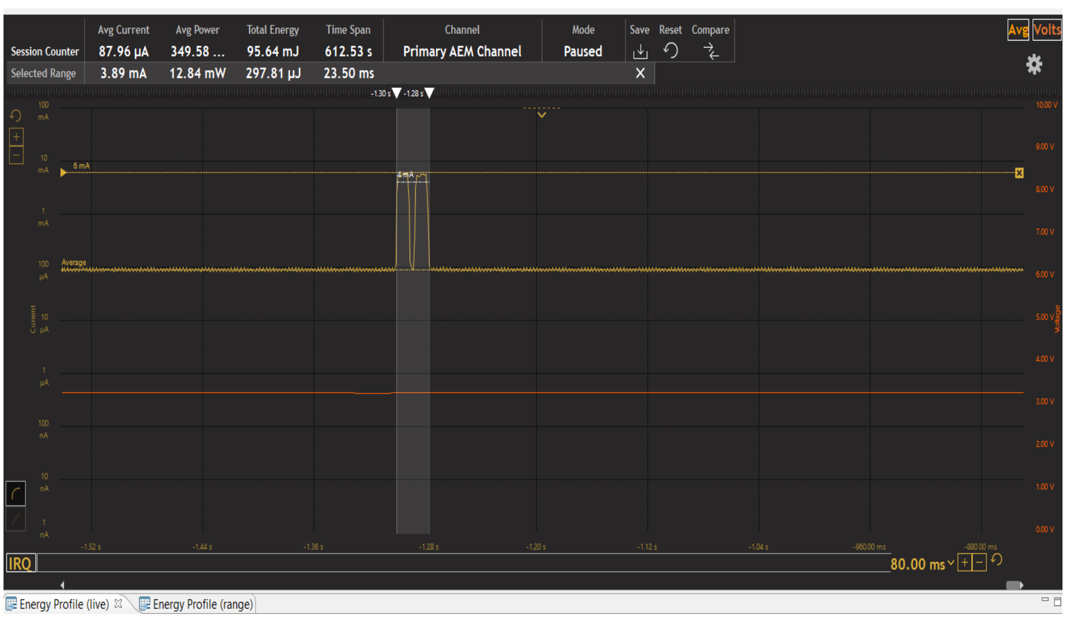

Fig. 7.13. Overall Power Consumption

7.4 Fall Pattern Test Results

7.4.1 Recognizing Patterns using Low-Pass and Complementary Filter Data

Figure 7.14 shows the pattern acquired during walking and standing using the pitch and roll angles from low-pass and complimentary filters. Figure 7.15 shows the data during rest and getting up from rest. There is a sudden rise and fall in `Comp_Roll` angle during the getting up position, while all the other angles decrease smoothly. Figure 7.16 shows all activity of daily living patterns. During walking and standing, there is smooth transition between the activities, but while performing sitting activity, there is a change in `Comp_Roll` and PitchAcc angle. Figure 7.17 shows the front fall pattern. During the fall a significant rise in PitchAcc, RollAcc, `Comp_Roll`, `Comp_Pitch` angles can be seen. Figure 7.18 shows the back-fall pattern. During back fall, `Comp_Roll` and PitchAcc angles rises, while RollAcc and `Comp_Pitch` angle decreases. Figure 7.19 and figure 7.20 shows the side fall patterns. The data pattern change can be seen during the right and left fall. Figure 7.21 shows the complete fall pattern.

7.4.2 Recognizing Fall Patterns using SVM and SMA

The data pattern acquired from low pass filter and complementary filter, exhibits sitting and resting activity pattern closely aligns with fall patterns. Therefore, adding one more layer of decision factor eliminates eliminate the risk of false fall detection. Figure 7.22 shows the SVM and SMA data for performing sitting, walking and standing activities. Figures 7.23, 7.24, 7.25 and 7.26 shows the fall pattern acquired.

Fig. 7.14. Data Pattern During Walking

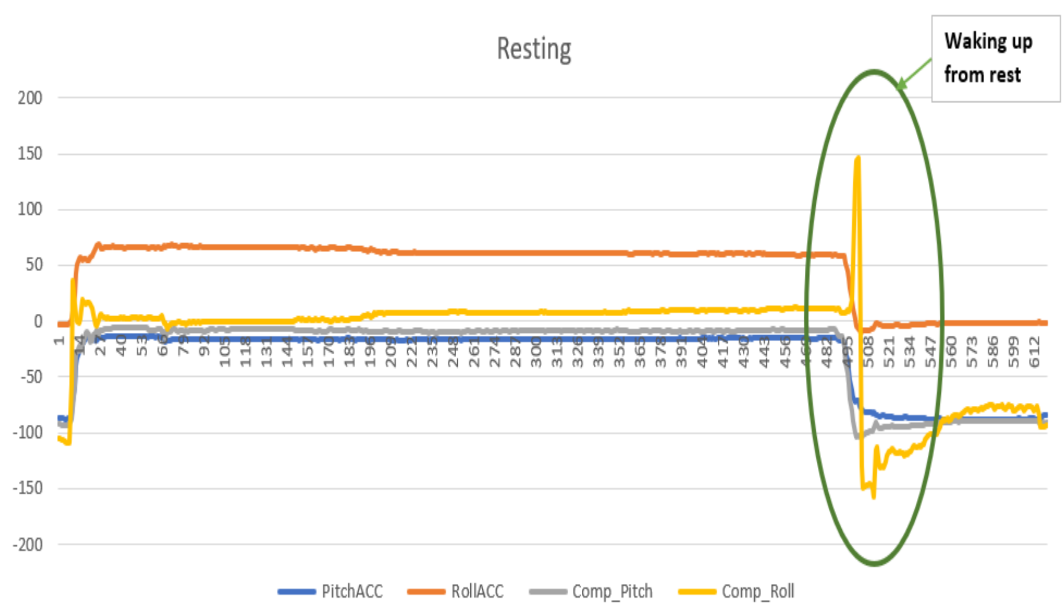

Fig. 7.15. Data Pattern During Resting

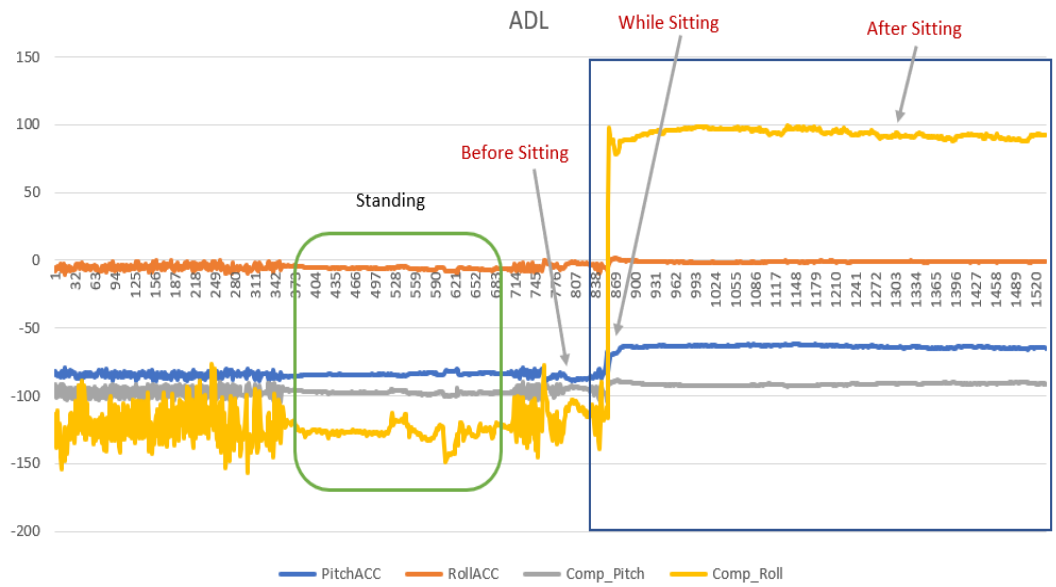

Fig. 7.16. Data Pattern During ADL

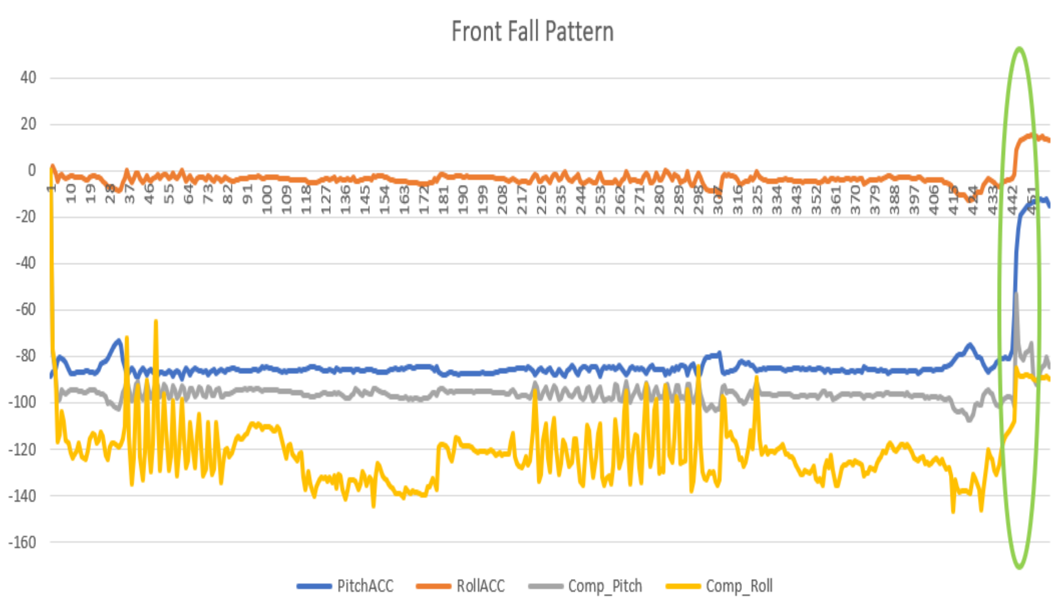

Fig. 7.17. Data Pattern During Front Fall

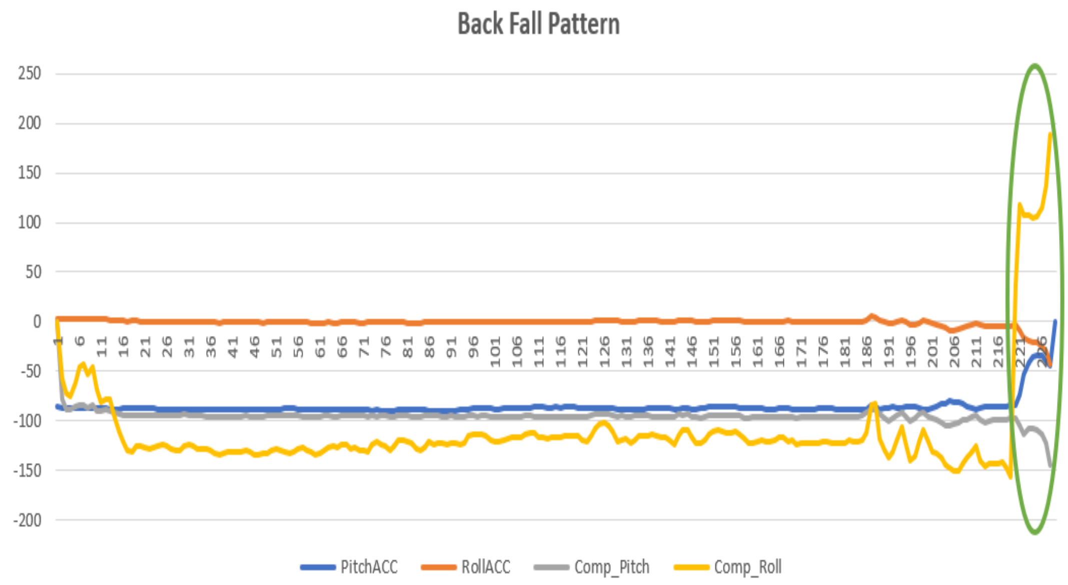

Fig. 7.18. Data Pattern During Back Fall

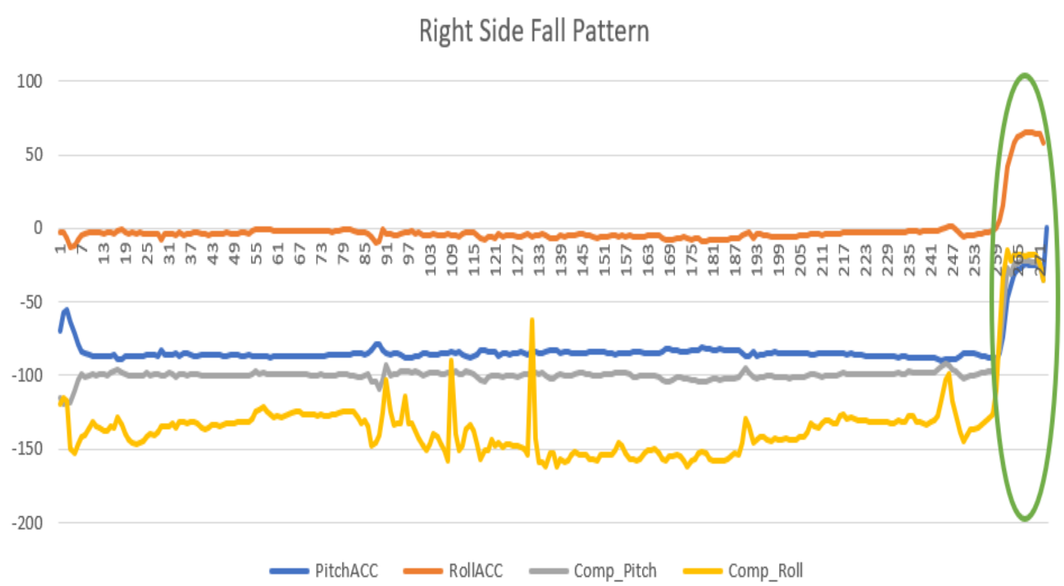

Fig. 7.19. Data Pattern During Right-Side Fall

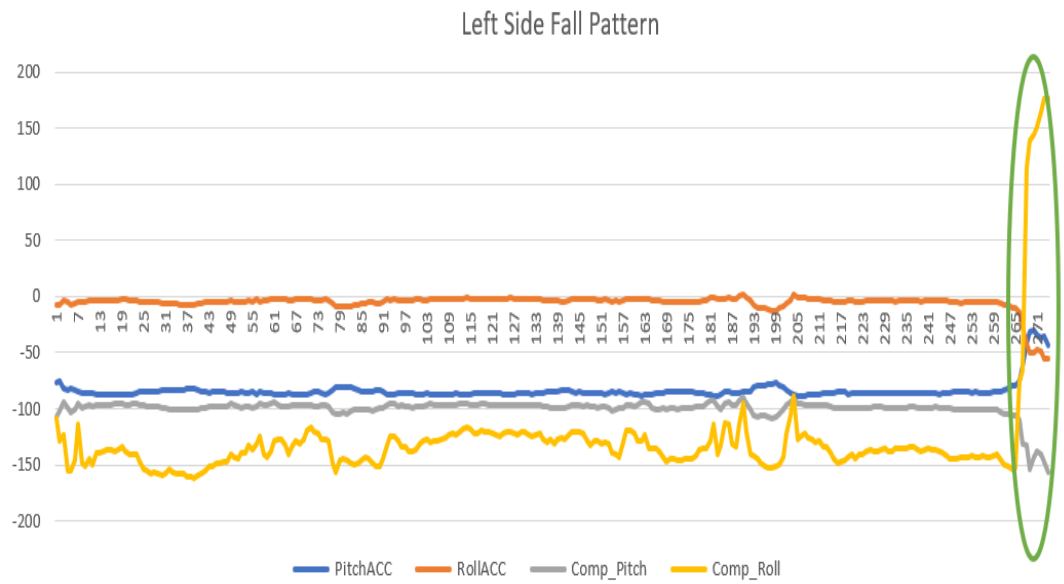

Fig. 7.20. Data Pattern During Left-Side Fall

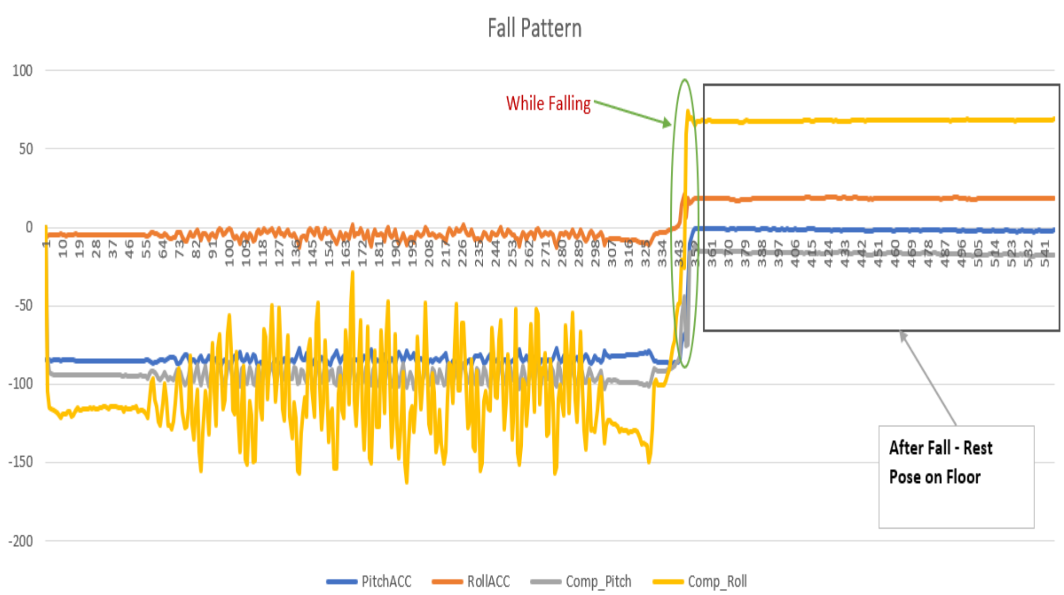

Fig. 7.21. Data Pattern During complete Fall

Fig. 7.22. ADL Data Pattern Using SVM and SMA

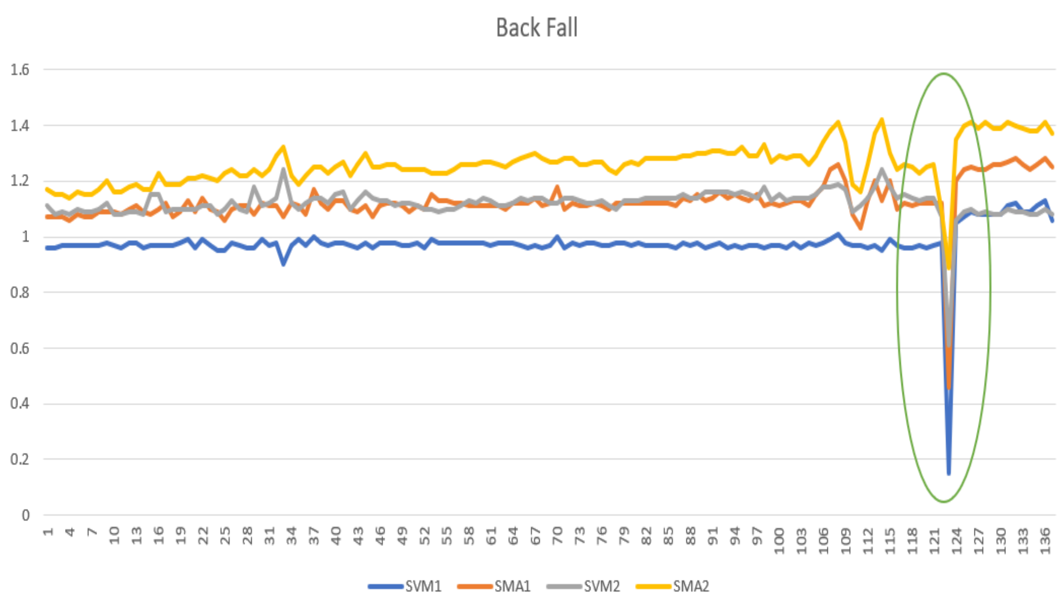

Fig. 7.23. Back-Fall Data Pattern Using SVM and SMA

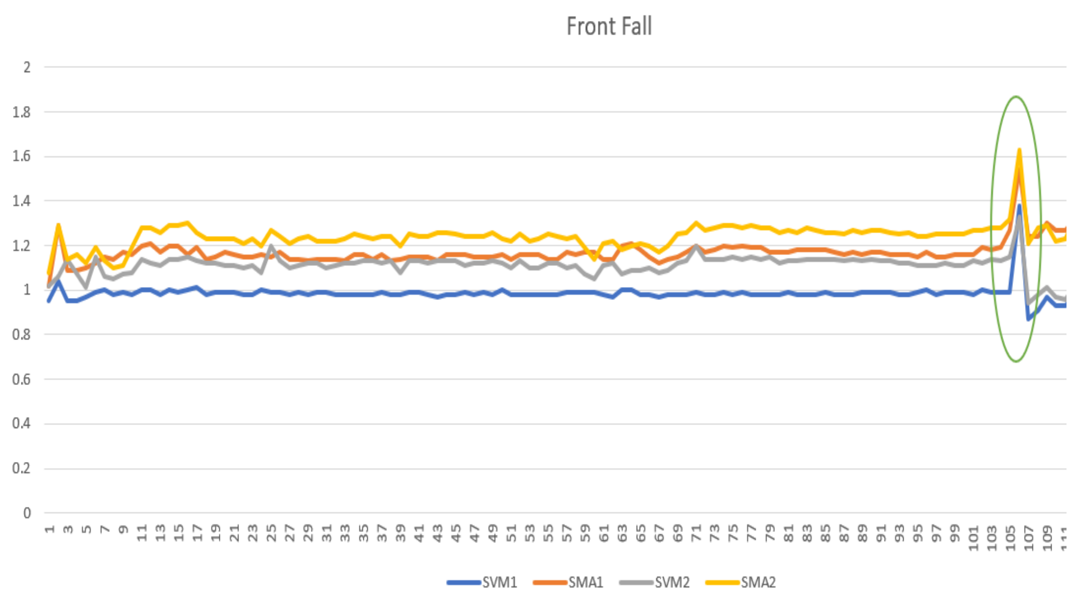

Fig. 7.24. Front-Fall Data Pattern Using SVM and SMA

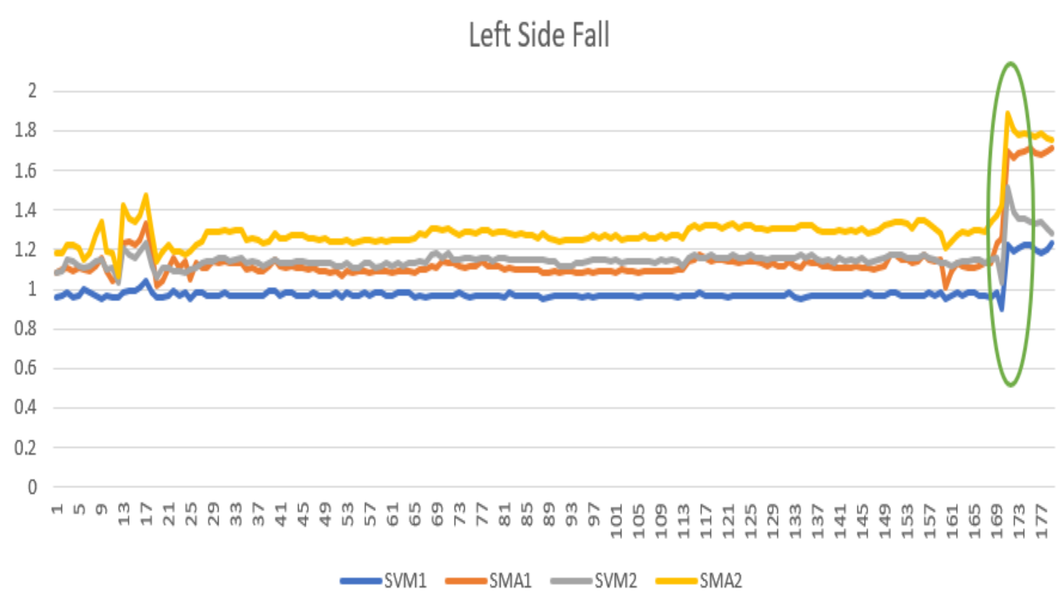

Fig. 7.25. Left-Side-Fall Data Pattern Using SVM and SMA

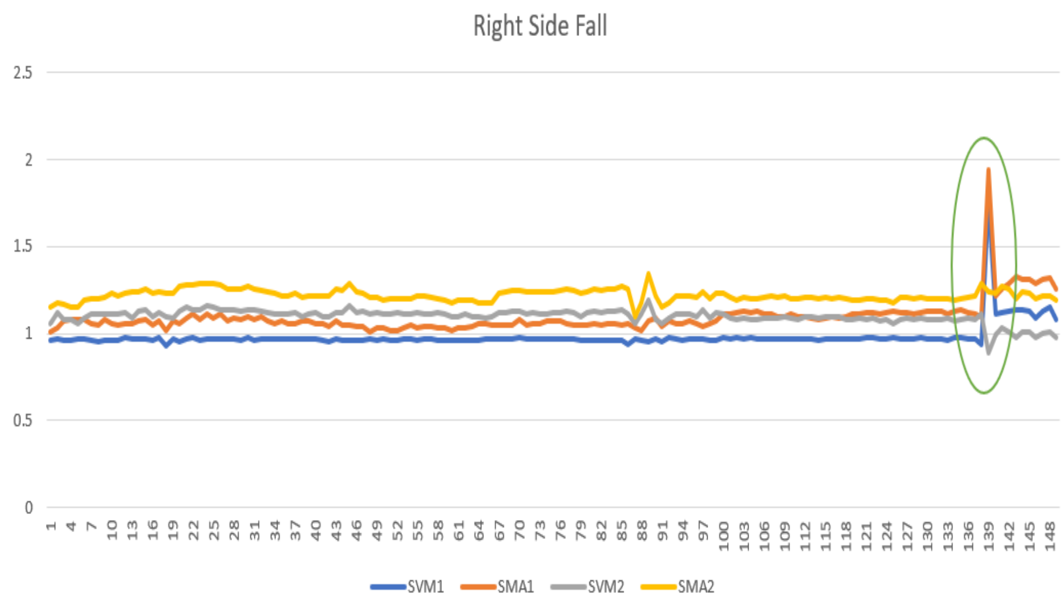

Fig. 7.26. Right-Side-Fall Data Pattern Using SVM and SMA

Table 7.1.
Data Set

Activity	**Incidents**	**True Positive Fall**	**False Negative Fall**
Human Fall	Front Fall	20	0
	Back Fall	15	0
	Left-side Fall	21	0
	Right-side Fall	21	0
ADL		**False Positive fall**	**True Negative fall**
		——	——
	Sitting Down	4	40
	Standing up	1	70
	Walking	0	70
	Resting	5	30

7.4.3 System Test Results

The results are obtained by performing test for four kinds of fall and four kinds of ADL. Each fall and ADL posture has been repeated at least 15 times on each subject to determine thresholds. The accuracy, sensitivity, and specificity is determined using equations 7.3, 7.1, 7.2 and from the test data in Table 7.1. The designed system estimates accuracy of 96.63%, sensitivity of 100%, and specificity of 95.45%.

$$Sensitivity = \frac{TPF}{TPF + FNF} \tag{7.1}$$

$$Specificity = \frac{TNF}{FPF + TNF} \tag{7.2}$$

$$Accuracy = \frac{TPF + TNF}{TPF + FNF + FPF + TNF} \tag{7.3}$$

8. CONCLUSION AND FUTURE WORK

In this work, a full characterization of the Pre-Fall detection system was conducted. This covered the hardware and software design, and the low power features. The system was simulated and built. The practical results showed significant impact on the speed and power consumption. Near 250ms was measured from the time of fall detection to the time when the fall occurs. This time is also analyzed to be sufficient to trigger safety devices that protect the patient when the fall occurs. Therefore, in the future work, the Pre-Fall prevention system can be designed and developed to prevent users knee, heap, and head areas from injuries. The comfort of the system can be further increased by making it a generic wearable device with low power IoT devices. Energy harvesting will be an important additive component to the system to elongate the battery life, and this is also reserved for future consideration. Within the approach, the subject's body postures were extracted from the angular velocity and acceleration data. The thresholding based classification model was used to differentiate between fall and ADL. The designed system results in accuracy of 96.63%, sensitivity of 100%, and specificity of 95.45%. The proposed design also features wireless capabilities to communicate fall information to caretakers.

The low power implementation was approached via the use of low power processor, the utilization of sleeping mode while in idle, and the software algorithm that featured low power. The low power is important in saving battery life, thus, minimizing system cost and increasing the reliability and the efficiency of the device.

The study also considered the optimum number of sensors and their locations on the body. The designed Pre-Fall detection system was found be optimum when worn by user on the waist.

REFERENCES

REFERENCES

[1] "Falls," 2018 (accessed 15 February, 2018). [Online]. Available: http://www.who.int/mediacentre/factsheets/fs344/en/

[2] D. Houry, C. Florence, G. Baldwin, J. Stevens, and R. McClure, "The cdc injury centers response to the growing public health problem of falls among older adults," *American journal of lifestyle medicine*, vol. 10, no. 1, pp. 74–77, 2016.

[3] E. R. Burns, J. A. Stevens, and R. Lee, "The direct costs of fatal and non-fatal falls among older adults united states," *Journal of safety research*, vol. 58, pp. 99–103, 2016.

[4] J. A. Stevens, P. S. Corso, E. A. Finkelstein, and T. R. Miller, "The costs of fatal and non-fatal falls among older adults," *Injury prevention*, vol. 12, no. 5, pp. 290–295, 2006.

[5] B. Vellas, F. Cayla, H. Bocquet, F. De Pemille, and J. Albarede, "Prospective study of restriction of activity in old people after falls," *Age and Ageing*, vol. 16, no. 3, pp. 189–193, 1987.

[6] D. Wild, U. Nayak, and B. Isaacs, "How dangerous are falls in old people at home?" *Br Med J (Clin Res Ed)*, vol. 282, no. 6260, pp. 266–268, 1981.

[7] "Costs of falls among older adults," 2016 (accessed 15 February, 2018). [Online]. Available: https://www.cdc.gov/homeandrecreationalsafety/falls/fallcost.html

[8] M. E. Tinetti and C. S. Williams, "Falls, injuries due to falls, and the risk of admission to a nursing home," *New England journal of medicine*, vol. 337, no. 18, pp. 1279–1284, 1997.

[9] A. Jefiza, E. Pramunanto, H. Boedinoegroho, and M. H. Purnomo, "Fall detection based on accelerometer and gyroscope using back propagation," in *Electrical Engineering, Computer Science and Informatics (EECSI), 2017 4th International Conference*. IEEE, 2017, pp. 1–6.

[10] S. Bhandari, N. Babar, P. Gupta, N. Shah, and S. Pujari, "A novel approach for fall detection in home environment," in *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, October 2017, pp. 1–5.

[11] M. Popescu and A. Mahnot, "Acoustic fall detection using one-class classifiers," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. IEEE, 2009, pp. 3505–3508.

[12] L. Ciabattoni, F. Ferracuti, G. Foresi, A. Freddi, A. Monteri, and D. P. Pagnotta, "Real-time fall detection system by using mobile robots in smart homes," in *2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, September 2017, pp. 15–16.

[13] P. Jatesiktat and W. T. Ang, "An elderly fall detection using a wrist-worn accelerometer and barometer," in *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE*. IEEE, 2017, pp. 125–130.

[14] Y. He, Y. Li, and S. D. Bao, "Fall detection by built-in tri-accelerometer of smartphone," in *Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics*, January 2012, pp. 184–187.

[15] F. Sposaro and G. Tyson, "ifall: an android application for fall monitoring and response," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. IEEE, 2009, pp. 6119–6122.

[16] R. Tiwari, A. K. Singh, and S. N. Khan, "Using android platform to detect free fall," in *2013 International Conference on Information Systems and Computer Networks*, March 2013, pp. 161–163.

[17] S. Gasparrini, E. Cippitelli, S. Spinsante, and E. Gambi, "A depth-based fall detection system using a kinect® sensor," *Sensors*, vol. 14, no. 2, pp. 2756–2775, 2014.

[18] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the microsoft kinect," *IEEE journal of biomedical and health informatics*, vol. 19, no. 1, pp. 290–301, 2015.

[19] L. Yang, Y. Ren, and W. Zhang, "3d depth image analysis for indoor fall detection of elderly people," *Digital Communications and Networks*, vol. 2, no. 1, pp. 24–34, 2016.

[20] E. Cippitelli, F. Fioranelli, E. Gambi, and S. Spinsante, "Radar and rgb-depth sensors for fall detection: a review," *IEEE Sensors Journal*, vol. 17, no. 12, pp. 3585–3604, 2017.

[21] G. Feng, J. Mai, Z. Ban, X. Guo, and G. Wang, "Floor pressure imaging for fall detection with fiber-optic sensors," *IEEE Pervasive Computing*, vol. 15, no. 2, pp. 40–47, 2016.

[22] N. Otanasap, "Pre-impact fall detection based on wearable device using dynamic threshold model," in *2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, December 2016, pp. 362–365.

[23] J. Liu and T. E. Lockhart, "Development and evaluation of a prior-to-impact fall event detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 7, pp. 2135–2140, 2014.

[24] A. Leone, G. Rescio, and P. Siciliano, "Fall risk evaluation by surface electromyography technology," *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pp. 1092–1095, 06 2017.

[25] S. Shan and T. Yuan, "A wearable pre-impact fall detector using feature selection and support vector machine," in *IEEE 10th International Conference on Signal Processing*, October 2010, pp. 1686–1689.

[26] G. Wu and S. Xue, "Portable preimpact fall detector with inertial sensors," *IEEE Transactions on neural systems and Rehabilitation Engineering*, vol. 16, no. 2, pp. 178–183, 2008.

[27] A. T. Özdemir and B. Barshan, "Detecting falls with wearable sensors using machine learning techniques," *Sensors*, vol. 14, no. 6, pp. 10 691–10 708, 2014.

[28] P. Rosero-Montalvo, D. Peluffo-Ordóñez, P. Godoy, K. Ponce, E. Rosero, C. Vásquez, F. Cuzme, S. Flores, and Z. Mera, "Elderly fall detection using data classification on a portable embedded system," in *Ecuador Technical Chapters Meeting (ETCM), 2017 IEEE*. IEEE, 2017, pp. 1–4.

[29] A. T. Özdemir, "An analysis on sensor locations of the human body for wearable fall detection devices: Principles and practice," *Sensors*, vol. 16, no. 8, p. 1161, 2016.

[30] S. C. Chavan and A. Chavan, "Smart wearable system for fall detection in elderly people using internet of things platform," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, June 2017, pp. 1135–1140.

[31] F. R. G. Cruz, M. P. Sejera, M. B. G. Bunnao, B. R. Jovellanos, P. L. C. Maao, and C. J. R. Santos, "Fall detection wearable device interconnected through zigbee network," in *2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, December 2017, pp. 1–6.

[32] L. Silicon, "Efm32gg:stk3700 user manual," *Rev.0.21*, 2013 (accessed 18 February, 2018). [Online]. Available: https://www.silabs.com/documents/public/user-guides/efm32gg-stk3700-ug.pdf

[33] Silicon Labs, "Efm32gg230 datasheet," *Rev.0.90*, 2011 (accessed 18 February, 2018). [Online]. Available: https://www.silabs.com/documents/public/data-sheets/EFM32GG230.pdf

[34] InvenSense, "Mpu-6050 datasheet," *Rev.3.4*, 2013 (accessed 15 February, 2018). [Online]. Available: https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf

[35] A. Devices, "Adxl345 3-axis digital accelerometer datasheet," *Rev.E*, 2015 (accessed 15 February, 2018). [Online]. Available: http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf

[36] T. Instruments, "Cc2541: 2.4-ghz bluetooth® low energy and proprietary system-on-chip," *Rev.D*, 2013 (accessed 18 February, 2018). [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2541.pdf

[37] F. Wu, H. Zhao, Y. Zhao, and H. Zhong, "Development of a wearable-sensor-based fall detection system," *International journal of telemedicine and applications*, vol. 2015, p. 2, 2015.

[38] H. G. Min and E. T. Jeung, "Complementary filter design for angle estimation using mems accelerometer and gyroscope," *Department of Control and Instrumentation, Changwon National University, Changwon, Korea*, pp. 641–773, 2015.

[39] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 156–167, January 2006.